

Bachelorarbeit 2014

One Laptop per Child *Mathematik Lernspiel*



Student : Kevin Kreuzer
Dozent : Laurent Sciboz
Abgabe : 25. Juli 2014
Webseite : www.hevs.ch

Zusammenfassung

Im Rahmen meiner Bachelorarbeit befasse ich mich mit der Konzeption und Implementierung einer Applikation für die non-profit Organisation „One Laptop per Child“ (OLPC). One Laptop per Child hat es sich zur Mission gemacht den ärmsten Kindern dieser Welt Zugang zu Bildung zu verschaffen. Um diese Mission zu verwirklichen wird jedem Kind ein extra angefertigter Laptop zur Verfügung gestellt. Dieser Laptop wurde speziell für die Anforderungen der dritten Welt entworfen. So kann er in extremer Hitze eingesetzt werden, ist wasserresistent und benötigt nur wenig Energie.

Ziel meiner Arbeit ist die Entwicklung und Umsetzung einer Applikation für ein OLPC Gerät. Mit Hilfe dieser Applikation können sich Kinder zwischen 6 und 12 Jahren Basiswissen aneignen. Den Kindern soll fundamentales Wissen mit auf den Weg gegeben werden. Wissen, auf dem die Kinder aufbauen können. Wissen, mit dem sie künftig in der Lage sind eigene Probleme zu lösen.

Die Arbeit wurde in der Zeitspanne vom 12. April 2014 bis zum 28. Juli 2014 realisiert. Bis zum 13. Juni 2014 konnten aufgrund anderer Module an der Fachhochschule wöchentlich lediglich drei Tage für die Bachelorarbeit verwendet werden. Die Zeit vom 09. Juni 2014 bis zum 20. Juni 2014 galt zudem der Prüfungsvorbereitung und den Semesterprüfungen. Die Arbeit wurde von Laurent Sciboz betreut.

Schlüsselbegriffe: OLPC, Lernsoftware, Sugar Web, Linux

Danksagung

Zunächst möchte ich mich bei allen Personen bedanken die mich während meiner Bachelorarbeit unterstützt und motiviert haben.

Besonders danken möchte ich Herrn Johannes Krampf der mir bei technischen Schwierigkeiten weiterhelfen konnte. Er half mir nicht nur bei der Beseitigung einiger Probleme sondern gab mir durch kritisches Hinterfragen immer wieder wertvolle Tipps.

Weiter möchte ich mich bei Fabian Cretton bedanken. Dieser hat mir bei generellen Fragen betreffend OLPC immer weitergeholfen oder mir die nötigen Nachschlagewerke sowie Kontaktadressen zur Verfügung gestellt.

Der Dank richtet sich auch an Christoph Guéret. Aufgrund seiner Nähe zum OLPC Projekt war er bei Unklarheiten immer eine hilfreiche Ansprechperson. Alle seiner prompten Rückmeldungen halfen mir stets weiter.

Für die Übersetzungen der Texte ins Spanische möchte ich mich bei María De Arteaga bedanken.

Darüber hinaus bin ich auch allen Personen welche mir während der Analysephase meiner Bachelorarbeit für Auskünfte und Informationen zur Verfügung standen dankbar.

Für die gute Betreuung meiner Bachelorarbeit danke ich Laurent Sciboz.

Inhaltsverzeichnis

1	Abbildungsverzeichnis	viii
2	Tabellenverzeichnis	x
3	Abkürzungsverzeichnis	xi
4	Einleitung	1
5	Methodik	2
5.1	Recherche- und Analysephase	2
5.2	Implementierungsphase	4
5.3	Testphase	4

Analyse

6	OLPC – One Laptop per Child	7
7	OLPC Geräte	7
7.1	OLPC XO Laptop	7
7.1.1	OLPC XO Laptop – Hardware	7
7.1.2	OLPC XO Laptop – Software	8
7.2	OLPC Tablet	9
7.2.1	OLPC Tablet – Hardware	9
7.2.2	OLPC Tablet - Software	10
8	Erster Entscheid - OLPC XO Laptop vs OLPC XO Tablet	10
9	OLPC Einsatzgebiete	11
10	Zweiter Entscheid – Welche Sprache und welche Länder?	15
11	Was wird gebraucht - Welches ist das Bedürfnis?	15
11.1	Subsahara-Afrika	19
11.2	Lateinamerika und Karibik	21
11.3	Süd- West Asien	24
11.4	Zentralasien	25
12	Dritter Entscheid – Welches Bedürfnis soll gedeckt werden	25
13	Technologien	27
13.1	Phyton	27
13.2	Sugar Web	27
14	Vierter Entscheid – Welche Technologie soll benutzt werden?	28
15	Fünfter Entscheid – Welche Entwicklungsumgebung wird gebraucht?	29
16	Die neue Applikation	30
16.1	State of the art - Liste der Webaktivitäten	30
16.2	Was wird im Bereich der Mathematik gebraucht?	33
16.2.1	Massachusetts Curriculum Framework for Mathematics	33
16.2.2	Operationen und algebraisches Denken	33
16.2.3	Nummern und Operationen der Zehnerbasis	34
16.2.4	Bruchrechnen	35
16.2.5	Messungen von Daten	35
16.2.6	Geometrie	36
16.3	Anforderungen des Massachusetts Curriculum Framework for Mathematics in Bezug of OLPC	37

16.3.1	Zahlenverständnis und Umgang mit Operationen	37
16.3.2	Muster/Modelle, Relationen und Algebra	39
16.3.3	Geometrie	40
16.3.4	Messungen	41
16.3.5	Datenanalyse, Statistik und Wahrscheinlichkeit	42
16.4	Konkrete Softwareideen	43
17	Sechster Entscheid - Bestimmen der Grundidee	44
18	Zusammenfassung der Entscheidungen	45

Das Spiel

19	Aventura Matemática	48
19.1	Anforderungen	48
19.1.1	Nichtfunktionale Musskriterien:	48
19.1.2	Funktionale Musskriterien:	49
19.1.3	Funktionale Kannkriterien:	49
19.2	Spielbeschrieb	49
19.3	Ablauf einer Spielrunde	50
19.4	Das Spiel beginnt!	51
19.5	Hauptaktivität	52
19.5.1	Kurzbeschrieb und Übersicht	52
19.5.2	Levels	52
19.5.2.1	Level 1 – Isla Matemática	52
19.5.2.2	Level 2 – Universo Matemático	53
19.5.2.3	Level 3 – Viejo Oeste Matemático	54
19.5.3	Use Case	54
19.5.3.1	Würfeln	54
19.6	Die Minispiele	56
19.6.1	Allgemeiner Überblick	56
19.6.2	Der MiniShop	59
19.6.2.1	Spielbeschrieb und Übersicht	59
19.6.2.2	Use Cases	60
19.6.3	Fisch-Spiel	61
19.6.3.1	Spielbeschrieb und Übersicht	61
19.6.3.2	Use Cases	62
19.6.4	Formen zählen	63
19.6.4.1	Spielbeschrieb und Übersicht	63
19.6.4.2	Use Cases	64
19.6.5	Plus und Minus einsetzen	65
19.6.5.1	Spielbeschrieb und Kurzübersicht	65
19.6.5.2	Use Cases	65
19.6.6	Muster mit Hilfe von Formen erkennen	66
19.6.6.1	Kurzbeschrieb und Übersicht	66
19.6.6.2	Use Cases	67
19.6.7	Sortiert Mini-Spiel	68
19.6.7.1	Kurzbeschrieb und Übersicht	68
19.6.7.2	Use Cases	69
19.6.8	Zehnerspiel Mini-Spiel	69
19.6.8.1	Use Cases	70
19.6.9	Wand Mini-Spiel	71
19.6.9.1	Kurzbeschrieb und Übersicht	71
19.6.9.2	Use Cases	71
19.6.10	Slider Mini-Spiel	72
19.6.10.1	Kurzbeschrieb und Übersicht	72
19.6.10.2	Use Cases	73

19.6.11	Vervollständige den Käse“ Mini-Spiel	74
19.6.11.1	Kurzbeschreibung und Übersicht	74
19.6.11.2	Use Cases	74
19.6.12	Winkel und Kreise	75
19.6.12.1	Kurzbeschreibung und Übersicht	75
19.6.12.2	Use Cases	76
19.6.13	Mathematiktafel	76
19.6.13.1	Kurzbeschreibung und Übersicht	76
19.6.13.2	Use Cases	77
19.6.14	Gitteradditionsspiel	78
19.6.14.1	Kurzbeschreibung und Übersicht	78
19.6.14.2	Use Cases	79
19.6.15	Memory-Spiel	80
19.6.15.1	Kurzbeschreibung und Übersicht	80
19.6.15.2	Use Cases	81
19.6.16	Schloss Mini-Spiel	81
19.6.16.1	Kurzbeschreibung und Übersicht	81
19.6.16.2	Use Cases	82
19.6.17	Teil Spiel	83
19.6.17.1	Kurzbeschreibung und Übersicht	83
19.6.17.2	Use Cases	84
19.7	Sonderfälle	84
19.7.1	Goldenes Feld – Besuch beim Affen	85
19.7.2	Schwarzes Feld - Tiger	86
19.7.2.1	Spielablauf	87
19.7.2.2	Use Cases	88

Technische Dokumentation

20	Vorbereitung des Systems	90
21	Verwendete Bibliotheken	93
21.1	D3 – Data-Driven Documents	93
21.2	jQuery	95
22	Single-page Web Applikation	96
23	Struktur	100
24	Multimedia Dateien	102
24.1	Bilder	102
24.2	Sound	102
25	Aventura Matemática auf dem OLPC Laptop	103
26	Git Version Control	105
27	Testing	106
28	Aktuelle Verfügbarkeit des Spiels	107
29	Abgabedokumente	107
30	Schlussfolgerung	109
31	Literaturverzeichnis	111
Anhang I: Interview mit Otto Imsand		113
Anhang II: Arbeitsjournal		116
Anhang III – Quellcode mainscript.js		121

Anhang IV – Quellcode Schlossspiel.....	147
Anhang V – Quellcode Memoryspiel.....	151
Anhang VI – Quellcode Quizspiel.....	155
Anhang VII – Quellcode „Was kommt als nächstes?“	158
32 Selbständigkeitserklärung	163

1 Abbildungsverzeichnis

Abbildung 1 : (Eigene Darstellung) Anzahl OLPC Laptops auf der Weltkarte eingefärbt. Quelle: (One Laptop Per Child)	13
Abbildung 2 : (Eigene Darstellung) Sprachen der Länder in denen OLPC Laptops eingesetzt werden. Quelle: (One Laptop Per Child)	14
Abbildung 3 : Kinder die nicht zur Schule gehen (Global). Quelle: (Unesco.org, 2014).....	15
Abbildung 4 : (Eigene Darstellung) Grafik zur Erreichung der EFA Ziele (Stand: 2013/2014) : Quelle (UNESCO, 2014).....	17
Abbildung 5 : (Eigene Darstellung) Lesefähigkeiten weltweit (Stand: 2013/2014). Quelle: (UNESCO, 2014)	18
Abbildung 6 : (Eigene Darstellung) Lesefähigkeiten Subsaharisches Afrika. Quelle: (UNESCO, 2014)	19
Abbildung 7 : (Eigene Darstellung) Subsahra-Afrika / Anzahl Kinder welche die Grundlagen der Mathematik beherrschen. Quelle: (UNESCO, 2014).....	21
Abbildung 8 : (Eigene Darstellung) Lesefähigkeit Südamerika. Quelle: (UNESCO, 2014)...	22
Abbildung 9 : (Eigene Darstellung) Lateinamerika und Karibik / Anzahl Kinder welche die Grundlagen der Mathematik beherrschen. Quelle: (UNESCO, 2014).....	23
Abbildung 10 : (Eigene Darstellung) Lesefähigkeit Süd-West Asien. Quelle: (UNESCO, 2014).....	24
Abbildung 11 : (Eigene Darstellung) Lesefähigkeit Zentralasien. Quelle: (UNESCO, 2014)	25
Abbildung 13 : (Eigene Darstellung) Brainstorming Teil 1	43
Abbildung 14 : (Eigene Darstellung) Brainstorming Teil 2	44
Abbildung 15 : (Eigene Darstellung) Use Case Adventura Mathematica	50
Abbildung 16 : (Eigene Darstellung) Sequenzdiagramm Ablauf Runde	51
Abbildung 17 : (Eigene Darstellung) Auswahlbildschirm / Spielbeginn.....	51
Abbildung 18 : (Eigene Darstellung) Screenshot Level 1	53
Abbildung 19 : (Eigene Darstellung) Screenshot Level 2.....	53
Abbildung 20 : (Eigene Darstellung) Screenshot Level 3.....	54
Abbildung 21 : (Eigene Darstellung) Use Case Hauptaktivität	54
Abbildung 22 : (Eigene Darstellung) Anzeige gewürfelte Zahl	55
Abbildung 23 : (Eigene Darstellung) Vorwärtsbewegung des Spielers.....	55
Abbildung 24 : (Eigene Darstellung) Minispiel Shop.....	59
Abbildung 25 : (Eigene Darstellung) Use Case Minispiel Shop.....	60
Abbildung 26 : (Eigene Darstellung) Graphische Schnittstelle Fischspiel	61
Abbildung 27 : (Eigene Darstellung) Use Case Fischspiel	62
Abbildung 28 : (Eigene Abbildung) Interface Spiel Formen zählen	64
Abbildung 29 : (Eigene Abbildung) Use Case Spiel Formen zählen.....	64
Abbildung 30 : (Eigene Darstellung) Grafische Oberfläche Endresultat Mini-Spiel	65
Abbildung 31 : (Eigene Darstellung) Use Case Endresultat Mini-Spiel.....	65
Abbildung 32 : (Eigene Darstellung) Grafische Schnittstelle „Was kommt als nächstes?“	66
Abbildung 33 : (Eigene Darstellung) Use Case „Was kommt als nächstes?“	67
Abbildung 34 : (Eigene Darstellung) Beispiel eines Musters	67
Abbildung 35 : (Eigene Darstellung) Sortiert Mini-spiel Benutzeroberfläche	68
Abbildung 36 : (Eigene Darstellung) Use Case Sortiert Mini-Spiel	69
Abbildung 37 : (Eigene Darstellung) Zehnerspiel Benutzeroberfläche	70
Abbildung 38 : (Eigene Darstellung) Use Case Zehnerspiel	70
Abbildung 39 : (Eigene Darstellung) Graphische Benutzeroberfläche Wand Mini-Spiel.....	71
Abbildung 40 : (Eigene Darstellung) Use Case Diagramm Wand Mini-Spiel	71

Abbildung 41 : (Eigene Darstellung) Grafische Benutzeroberfläche Slider Mini-Spiel	72
Abbildung 42 : (Eigene Darstellung) Use Case Slidergame	73
Abbildung 43 : (Eigene Darstellung) Käsespiel Benutzeroberfläche	74
Abbildung 44 : (Eigene Darstellung) Use Case Diagramm Käse Spiel	74
Abbildung 45: (Eigene Darstellung) Benutzeroberfläche Winkel und Kreise	75
Abbildung 46 : (Eigene Darstellung) Use Case Winkel und Kreise	76
Abbildung 47 : (Eigene Darstellung) Benutzeroberfläche Mathtable	77
Abbildung 48 : (Eigene Darstellung) Use Case Diagramm Math Table	77
Abbildung 49 : (Eigene Darstellung) Benutzeroberfläche Gitteraditionsspiel vor Auswahl...	78
Abbildung 50 : (Eigene Darstellung) Benutzeroberfläche Gitteraditionsspiel nach Auswahl.	79
Abbildung 51 : (Eigene Darstellung) Use Case Gitteraditionsspiel	79
Abbildung 52 : (Eigene Darstellung) Benutzeroberfläche Memory-Spiel	80
Abbildung 53 : (Eigene Darstellung) Use Case Memory-Spiel	81
Abbildung 54 : (Eigene Darstellung) Benutzeroberfläche Schloss Mini-Spiel	82
Abbildung 55 : (Eigene Darstellung) Use Case Castlegame	82
Abbildung 56 : (Eigene Darstellung) Benutzeroberfläche Teil Spiel	83
Abbildung 57 : (Eigene Darstellung) Use Case Teil Spiel	84
Abbildung 58 : (Eigene Darstellung) Aktivitätsdiagramm – Bananenabgabe	85
Abbildung 59 : (Eigene Darstellung) Abgabe der Bananen Level 3	86
Abbildung 60 : (Eigene Darstellung) Quizgame Eingabe	87
Abbildung 61 : (Eigene Darstellung) Quizgame nach Eingabe	88
Abbildung 62 : (Eigene Darstellung) Use Case Quizgame	88
Abbildung 64 : (Eigene Darstellung) Felddaten Level 1	94
Abbildung 65 : (Eigene Darstellung) Elemente auf dem Spielbrett zeichnen	95
Abbildung 67 : (Eigene Darstellung) Codebeispiel Kendo	96
Abbildung 68 : (Eigene Darstellung) Sequenzdiagramm einzelne HTML Seite	97
Abbildung 69 : (Eigene Darstellung) Methode Hide()	98
Abbildung 70 : (Eigene Darstellung) Methode Draw und Delete	98
Abbildung 71 : (Eigene Darstellung) Method Checkfield im Manuscript	99
Abbildung 72 : (Eigene Darstellung) Resultatanzeige	100
Abbildung 73 : (Eigene Darstellung) Ordnerstruktur Journeygame	100
Abbildung 75 : (Eigene Darstellung) Aventura Matematica unter osbuild	105
Abbildung 76 : Git Statistiken. Quelle: (Github.com, 2014)	106

2 Tabellenverzeichnis

Tabelle 1 : (Eigene Darstellung) Hardwarekomponenten OLPC Laptop. Quelle: (One Laptop per Child).....	8
Tabelle 2 : (Eigene Darstellung) Hardwarekomponenten OLPC Tablet. Quelle: (One Laptop per Child).....	10
Tabelle 3 : (Eigene Darstellung) Verteilung OLPC Laptop. Quelle: (One Laptop Per Child)	11
Tabelle 4 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Zahlenverständnis und Basisoperationen. Quelle: (Wiki Sugarlabs, 2014)	37
Tabelle 5 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Muster / Modelle, Relationen und Algebra. Quelle: (Wiki Sugarlabs, 2014)	39
Tabelle 6 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Muster / Geometrie. Quelle: (Wiki Sugarlabs, 2014).....	40
Tabelle 7 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Messungen. Quelle: (Wiki Sugarlabs, 2014).....	41
Tabelle 8 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Datenanalyse, Statistik und Wahrscheinlichkeit. Quelle: (Wiki Sugarlabs, 2014).....	42
Tabelle 9 : (Eigene Darstellung) Mini-Spiele und die dazugehörige Farbe.....	56
Tabelle 10 : (Eigene Darstellung) Nutzen-Wert der Mini-Spiele	57
Tabelle 11 : (Eigene Darstellung) Vorkommnisse der Mini-Spiele in den Levels	58
Tabelle 12 : (Eigene Darstellung) Fischspiel - Vergabe der Bananen	63

3 Abkürzungsverzeichnis

OLPC	One Laptop per Child
GTK	Gimp Toolkit
UNDP	United Nations Development Programm
GNU GPL	GNU General Public License
FOSS	Free Libre Open Source Software
HTML	Hyper Text Markup Language
DANS	Data Archiving and Network Services
CSS	Cascading Style Sheet
TIMSS	Trends in International Mathematics and Science Study
PIRLS	Progress in International Reading Literacy Study
SACMEQ	Southern and Eastern Africa Consortium for Monitoring Educational Quality
PASEC	Programme on the Analysis of Education Systems of the CONFEMEN
SERCE	Second Regional Coparative and Explanatory Study
DEZA	Direktion für Entwicklung und Zusammenarbeit
API	Application Programming Interface
RAD	Rapid Application Development
DOM	Document Object Model
JSON	JavaScript Object Notation
W-LAN	Wireless Local Area Network
VPN	Virtual Private Network
IIS	Internet Information Services
MB	Megabyte
KB	Kilobyte
MHz	Megahertz

4 Einleitung

In unserem Umfeld hat wohl jeder eine Grundausbildung abgeschlossen. Was für uns allerdings als selbstverständlich und normal gilt ist für so manche Kinder ein großes Privileg. Weltweit haben rund 72 Millionen keine Möglichkeit eine Grundschule zu besuchen. Der Großteil dieser Kinder sind Mädchen. Besonders für Kinder ethnischer Minderheiten, benachteiligter Völkergruppen oder Kinder welche in Armut oder in einer Konflikt-Krisenregion leben, ist der Zugang zu Bildung besonders schwierig. Hinzu kommt, dass nicht selten Kinder die Grundschule abbrechen. Daher überrascht es nicht, dass weltweit rund 759 Millionen Menschen (über 15 Jahre) nicht lesen können. Gemäß Prognosen des Weltbildungsberichts wird sich die Situation künftig aufgrund fehlender finanzieller Mittel und der Wirtschaftskrise noch verschlechtern. Doch nicht nur der Zugang zu Bildung allein ist problematisch. Auch die Qualität des Unterrichts ist ungenügend. Trotz Grundschulbildung beherrschen viele Schüler kein Grundrechnen. Die Lehrpläne beinhalten keine Ziele, sind fachlich überladen und entsprechen oft nicht den Lernbedürfnissen der Grundschüler. Die Unterrichtsinhalte orientieren sich zudem oft viel zu wenig an der Lebenssituation der Kinder. (BMZ - Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung), (www.spiegel.de, 2014)

Warum in Bildung investieren? Damit in Bildung investiert werden kann müssen selbstverständlich zuerst die Grundbedürfnisse gedeckt werden. Politische Stabilität, ausreichend Nahrung und Frieden haben klare Priorität. Allerdings kann man diese Probleme nur kurzfristig lösen. Um eine nachhaltige Wirkung erzielen muss in Bildung investiert werden. Gemäß der Zeitung Zeit Online wird bereits im Weltentwicklungsbericht aus dem Jahr 1982 darauf hingewiesen, dass Bauern welche eine mindestens sechs jährige Grundschulbildung vorweisen, bis zu 25 Prozent höhere Erträge erzielen. Doch nicht nur im Kampf gegen die Armut ist Bildung ein wichtiges Mittel. Frauen mit Bildung werden seltener ungewollt schwanger. Besonders die Fähigkeit zu lesen ist fundamental. Wer sich diese Fähigkeit aneignet hat, kann sich jederzeit neue Informationen in Form von Medien beschaffen. Nebst Schule müssen selbstverständlich auch die sozialen und wirtschaftlichen Bedingungen stimmen. Es ist aber allgemein bekannt, dass Qualifikationsmangel und Armut zusammenhängen. Um dem Mangel an Bildung entgegenzuwirken ist es von höchster Priorität, dass man bereits Kinder im Grundschulalter fördert. (www.zeit.de, 1986), (Hinz, 2014).

5 Methodik

Dieser Abschnitt befasst sich mit dem methodischen Vorgehen meiner Bachelorarbeit. Mein Vorgehen lässt folgende Gliederung zu:

1. Recherche- und Analysephase
2. Implementierungsphase
3. Testphase

5.1 Recherche- und Analysephase

Bei meiner Bachelorarbeit ist die Recherchephase von hoher Priorität. In dieser Phase entscheide ich welche Art von Lernprogramm ich implementiere und auf welchem Gerät dieses Lernprogramm laufen soll. Die Recherche bildet die Grundlage für die Konzeption des Programmes und sämtlicher grundlegender Entscheidungen. Weiter informiere ich mich über die Technologien auf die OLPC in Zukunft setzt.

Zu Beginn meiner Arbeit vergleiche ich zuerst das Tablet und den OLPC Laptop. Im Anschluss liste ich alle Länder in denen die OLPC Laptops verteilt sind auf. Um einen besseren Überblick zu erhalten habe ich Länder anhand der Anzahl OLPC Laptops auf der Weltkarte eingefärbt. Da die Sprache sehr wichtig ist und daher besonders berücksichtigt werden muss, habe ich die Amtssprachen aller Länder herausgesucht. Dadurch konnte ich die verschiedenen Sprachen mit der Anzahl Laptops der jeweiligen Länder abgleichen. Zum Schluss habe ich diese Resultate mit Hilfe eines Kuchendiagramms prozentual visualisiert.

Da ich aus der technischen Richtung komme habe ich nur geringfügige Erfahrungen mit der Pädagogik. Es gilt herauszufinden in welchen Ländern welche Probleme in Bezug auf die Bildung anzutreffen sind. Um mich der Thematik zu nähern begann ich mit einer gründlichen Internetrecherche über OLPC. Diverse Bücher aus Bibliotheken halfen mir mich den Lernprozessen der Kinder vertraut zu machen. Um genau nachzuvollziehen welches die Bedürfnisse der Kinder in Bezug auf Bildung sind, nahm ich Kontakt mit verschiedensten Hilfsorganisationen auf. Darunter UNICEF und UNESCO. Leider erhielt ich hier nur allgemeine Informationen welche meine Fragen nicht beantworten konnten. Ich suchte also

den direkten Kontakt zu Lehrern von Schulen in Dritt-Welt- Ländern. Ich konnte sowohl mit Otto Imsand und Hopeville.ch Kontakte knüpfen.

Otto Imsand ist Lehrer an der Sekundarschule in Naters. Er ist in Naters geboren und dort aufgewachsen. Er besuchte zuerst das Lehrerseminar in Sitten und schloss später die Universität in Freiburg ab. Danach sammelte er einige Jahre Berufserfahrung in der Schweiz. Im Anschluss folgte dann ein neuer Lebensabschnitt. Otto Imsand wanderte nach Lima aus. Hier unterrichtete er 17 Jahre lang am Colegio Pestalozi. Ich hatte die Gelegenheit Otto Imsand persönlich zu treffen und ihm in einem Interview meine Fragen zu stellen welche er zusätzlich direkt an die Partnerschule in Lima weiterleitete. Durch das Interview und die Antwort konnte ich weitere interessante und aufschlussreiche Informationen gewinnen.

Hopeville.ch ist ein Oberwalliser Verein der die Menschen in Nigeria unterstützt. Aktuell beteiligt sich der Verein am Ausbau des Swiss Foundation Hospitals. Weiter legt man den Fokus auf den Bildungsbereich. Mitglieder des Vereins Hoepville.ch haben bereits in Nigeria vor Ort unterrichtet und konnten mir deshalb gezielt Auskunft über die Bedürfnisse der Kinder geben.

An der Sitzung vom 28.04.2014 hat mir Herr Laurent Sciboz vorgeschlagen zusätzlich noch die Direktion für Entwicklung und Zusammenarbeit (DEZA) zu kontaktieren. Durch den Kontakt mit der DEZA erhielt ich einige interessante Informationen und Dokumente. Unter anderem den aktuellsten Weltbildungsbericht. Ich hatte nun genügend Materialien gesammelt um die Frage nach den Bedürfnissen der Kinder in Dritt-Welt-Ländern zu beantworten.

Im umfassenden Weltbildungsbericht der UNICEF stieß ich auf einige interessante Zahlen und Statistiken. Aus diesem Weltbildungsbericht pickte ich die für meine Arbeit relevanten Informationen heraus. Diese Daten glich ich mit den Ländern in denen OLPC Laptops vorkommen ab, so dass ich nur die für die Arbeit relevanten Daten verwende. Nach einigen Berechnungen konnte ich die Daten mit Hilfe diverser Balkendiagramme visualisieren. Diese Diagramme sind sehr aufschlussreich und geben schnell Auskunft über die Bedürfnisse und Probleme der Kinder in Dritt-Welt-Ländern. Anhand von Interviews, Zeitungsberichten, Büchern, Webseiten Telefonaten und den Angaben von Hopeville und der DEZA konnte ich entscheiden welches Bedürfnis meine Applikation decken soll. Weiter verglich ich die

verschiedenen Technologien mit der die Applikation implementiert werden kann und wählte eine Technologie aus. Durch ein Brainstorming sammelte ich verschiedenste Softwareideen. Ich verglich die verschiedenen Ideen miteinander und konnte so einen endgültigen Entscheid für ein Programm fällen.

5.2 Implementierungsphase

Zu Beginn dieser Phase installierte ich die ganze Entwicklungsumgebung. Dazu habe ich auf meinem Computer ein Linux Fedora als virtuelle Maschine aufgesetzt und darauf mit osbuild eine volle Sugar Umgebung eingerichtet. Nach einigen ersten kleinen Programmen und Tutorials begann ich mit meiner Arbeit. Im größten Teil dieser Phase befasste ich mich mit der reinen Implementierung. Dabei fanden auch sämtliche Modellierungen und Dokumentationen der Spiele in dieser Phase statt. Da ich in meiner Arbeit verschiedenste Bibliotheken verwende war es notwendig mich während der Implementierungsphase mit neuen Technologien vertraut zu machen.

5.3 Testphase

Diese Phase lief teilweise parallel mit der Implementierungsphase. Nachdem ich eine neue Funktion entwickelt hatte wurde diese jedes Mal manuell getestet. Die Applikation wurde in Sugarizer und in osbuild getestet. An einem gewissen Projektstand befasste ich mich mit dem Gedanken die Applikation auf dem OLPC Laptop zu testen obwohl hierzu noch ein Update fehlte. Gemeinsam mit Herrn Johannes Krampf erarbeitete ich die Lösung mein Spiel auf einem Webserver abzulegen und mit Hilfe des auf dem OLPC Laptop integrierten Browsers darauf zuzugreifen. Um dies zu bewerkstelligen habe ich mir vom Informatikdienst der Fachhochschule Wallis eine Windows Server in Form einer virtuellen Maschine geben lassen. Auf diesem Windows Server habe ich die nötigen Rollen installiert und diesen als Webserver konfiguriert. Mit Hilfe dieses Webservers konnte ich nun mein Spiel auf dem OLPC Laptop starten. Als ich das Spiel auf dem OLPC Laptop testete merkte ich, dass ich gewisse Anpassungen vornehmen musste. Aufgrund der begrenzten Leistung wurden nicht alle Bilder übermittelt und nicht alle Sounds abgespielt. Mit Hilfe diverser Tools und Techniken gelang es mir dieses Problem zu lösen. Somit hatte ich nun eine Testumgebung aufgesetzt um die Applikation unter realen Bedingungen zu testen. Allerdings funktionierte dies nicht in allen Netzwerken. Um die Testumgebung noch stabiler zu machen stellte mir Herr Johannes Krampf in Zusammenarbeit mit dem Informatikdienst einen Router zur Verfügung. Mit Hilfe

dieses Routers konnte ich mein eigenes kleines Netzwerk einrichten und so eine stabile Testumgebung für meine Entwicklungen herstellen. Damit das Spiel auch von außerhalb zugänglich ist hat Herr Krampf eine Version des Spiels auf einem Webserver des Technopols abgelegt. Somit ist das Spiel mit einer Internetverbindung weltweit zugänglich.

Analyse

Dieser Teil befasst sich mit sämtlichen Analysen
und den grundlegenden Entscheidungsfindungen meiner Arbeit.

6 OLPC – One Laptop per Child

OLPC ist eine non-profit Organisation mit der Mission den weltweit ärmsten Kindern Zugang zu Bildung zu verschaffen. Dazu entwickelt OLPC speziell angefertigte Laptops und Tablets. Diese Geräte trotzen hohen Temperaturen, sind wasserresistent und darüberhinaus sehr robust. Selbstverständlich sind diese Geräte sehr günstig und brauchen nur wenig Energie. Doch nicht nur Hardware wird zur Verfügung gestellt. Die Geräte von OLPC können mit diversen Lernprogrammen ausgestattet werden. Ziel dieser Programme ist unterhaltsames und selbstständiges Lernen für Kinder. Die OLPC Laptops werden heute von 2 Millionen Kindern und Lehrern in 42 verschiedenen Ländern täglich gebraucht. Mit Hilfe dieser Laptops haben die Kinder die Möglichkeit zu lernen sowie Lernerfolge und gelerntes zu teilen. (One Laptop Per Child, 2014)

7 OLPC Geräte

Alle Angaben in diesem neunten Kapitel sind sinnngemäße Wiedergaben folgender Quellen: (One Laptop per Child) , (Popsci.com, 2014) und (wikipedia.org, 2014).

Seit 2005 stellt One Laptop per Child mit dem United Nations Development Programm (UNDP) günstige Laptops mit geringem Verbrauch für Drittwelt Kinder her. Im Herbst des vergangenen Jahres veröffentlichte OLPC ein neues Gerät das OLPC Tablet. Für meine Bachelorarbeit ist es wichtig diese beiden Geräte miteinander zu vergleichen und zu bestimmen welches wohl die besseren Zukunftsaussichten und den besseren Nutzen für Kinder in Drittwelt Ländern hat. Dieser Vergleich bestimmt zugleich für welches Gerät die Lernapplikation anschließend implementiert wird. (wikipedia.org, 2014)

7.1 OLPC XO Laptop

7.1.1 OLPC XO Laptop – Hardware

Der XO Laptop wurde von Akademie- und Industrieexperten in Zusammenarbeit entwickelt. Extreme Umweltbedingungen wie hohe Hitze und hohe Feuchtigkeit wurden dabei speziell berücksichtigt. Der XO hält daher Temperaturen bis zu 60° stand. Weiter wurden die OLPC Laptops so gebaut, dass diese im Falle einer Beschädigung selbst von Kindern eigenständig repariert werden können. Der XO Laptop ist beständig, energieeffizient und unterhaltsam. Für

Kinder ist er aufgrund seiner Form (Tragegriff) besonders leicht zu transportieren. Der XO Laptop steht für Konnektivität. So können sich beispielsweise Kinder welche draußen unter einem Baum arbeiten ohne zusätzliche Hardware miteinander verbinden. Das neuste Modell ist der XO-4 Touch. (One Laptop Per Child)

Dieser beinhaltet folgende Komponenten:

Tabelle 1 : (Eigene Darstellung) Hardwarekomponenten OLPC Laptop. Quelle: (One Laptop per Child)

RAM	1-2 GB
Interner Speicher	8 GB SSD (Solid-State-Transfer)
Festplatte	-
Interne Kabel	2
Dimension	242mm×228mm×32mm
CPU	Dual core ARM PJ4, 1 GHz or 1.2 GHz
Auflösung	Resolution: 1200 × 900 (200 DPI)
Wireless / Netzwerk	Integrierte IEEE 802.11/b/g (2.4 GHz) Wireless Netzwerk Schnittstelle. Ad-hoc und AP Modus Netzwerk unterstützt. Bluetooth unterstützt (nur mit optionalem IEEE 802.11a/b/g/n).

7.1.2 OLPC XO Laptop – Software

Als Betriebssystem wird die Linux Distribution Fedora installiert. Diese Distribution wurde vom Hersteller Red Hat auf besonders intelligenten Ressourcen und geringen Stromverbrauch optimiert. Bitfrost sorgt für die nötige Sicherheit auf einem OLPC Laptop. Diese Sicherheitsplattform vereinigt einige erfolgreich getestete Sicherheitskonzepte und soll somit in der Lage sein Viren und andere Malware abzuwehren. (One Laptop Per Child)

Bei der Benutzeroberfläche eines OLPC Laptops handelt es sich um eine für den OLPC maßgeschneiderte Oberfläche namens Sugar. Diese Benutzerfläche ist genau auf den End User eines OLPC Laptops angepasst. Für die Benutzung eines OLPC werden keine

Kenntnisse in Lesen, Schreiben oder Informatik vorausgesetzt. Die Computerbenutzung erfolgt durch selbsterklärende Symbole. In Sugar spricht man nicht von Anwendungen sondern von Aktivitäten. Ein Nutzer kann eine Aktivität entweder für sich selbst oder für einen bestimmten Personenkreis starten. So können Aktivitäten gemeinsam benutzt werden. Um den Lernprozess zu unterstützen vereinigt Sugar verschiedenste Konzepte der Kognitionspsychologie. Nach Beendigung der Schulausbildung soll Sugar durch GNOME ersetzt werden und den XO Laptop somit auch für berufliche Zwecke einsetzbar machen. Der XO Laptop beinhaltet bereits einige vorinstallierte Anwendungen. (wikipedia.org, 2014)

Zu diesen Anwendungen zählen:

- Webbrowser – Verwendet zur Darstellung die Gecko-Engine
- AbiWord – Textverarbeitungsprogramm (ermöglicht die parallele Echtzeitbearbeitung von Dokumenten)
- Eine Anwendung zum E-Mail Versand
- Ein Chat Programm mit Videokonferenz Funktion
- TamTam – ein Musiksyntheser zur gemeinsamen Erstellung von Musikprojekten
- Musik Memory – ein Memory Spiel mit Tönen statt Musik
- Ein Mahlprogramm für Kinder
- Squeak – Auf dem Laptop werden dazu EToys installiert. Dadurch können Spiele Klassiker wie Tetris oder auch SimCity auf den XO übertragen werden.

Quelle: (wikipedia.org, 2014)

7.2 OLPC Tablet

7.2.1 OLPC Tablet – Hardware

Im Sommer des vergangenen Jahres brachte OLPC das XO Tablet auf den Markt. Das Tablet selbst unterscheidet sich äußerlich vor allem aufgrund seiner auffälligen Schutzhülle von den herkömmlichen Tablets. (Popsci.com, 2014)

Das OLPC Tablet beinhaltet folgende Komponenten:

Tabelle 2 : (Eigene Darstellung) Hardwarekomponenten OLPC Tablet. Quelle: (One Laptop per Child)

RAM	1 GB
Interner Speicher	8 GB
Dimension	1024x600
Wireless / Netzwerk	Wi-Fi 802.11 b/g/n connectivity
Sonstiges	Integrierte Elternkontrolle

7.2.2 OLPC Tablet - Software

Das OLPC Tablet läuft auf einem Android 4.1 Betriebssystem. Gemeinsam mit Vivitar / Sakar und anderen Partnern hat OLPC eine ganz spezielle grafische Benutzeroberfläche kreiert. Diese grafische Schnittstelle zeigt verschiedenste Symbole mit Traumberufen. Das Tablet verfügt über 150 bereits vorinstallierte Applikationen von denen jede einem Traumberuf zugeordnet ist. Die Applikation die das Kind benutzt hängt von der Wahl des Traumberufs auf dem Startbildschirm ab. Aktuell ist das XO Tablet in Spanisch und Englisch verfügbar. (Popsci.com, 2014)

8 Erster Entscheid - OLPC XO Laptop vs OLPC XO Tablet

Ziel meiner Bachelorarbeit ist es eine Lernsoftware für arme Kinder in Drittwelt oder Schwellenländer für ein OLPC Gerät zu implementieren. Da OLPC sowohl ein Tablet als auch einen Laptop entwickelt hat gilt es eines dieser Geräte in Berücksichtigung der Spezifikation und Ziel der Arbeit mit Hilfe obenstehender Erläuterungen auszuwählen. Hierbei fällt die Entscheidung auf den OLPC XO Laptop (XO-4).

Bei meiner Entscheidung stütze ich mich auf folgende Argumente. Während sich die Hardware des Tablets nur anhand von einer Schutzhülle von einem herkömmlichen Tablet unterscheidet wurde der XO Laptop ganz anders konstruiert. Bei der Entwicklung wurden Umweltbedingungen die deren in Afrika, Südamerika oder Asien entsprechen berücksichtigt. Der XO Laptop wurde so konstruiert das er extremen Temperaturen standhält, Wasserresistent ist, leicht tragbar und leicht zu reparieren ist. Er ist viel robuster als das Tablet und ist im

Bezug auf die Hardware genau auf die Bedingungen der Einsatzgebiete abgestimmt. Für Kinder in Entwicklungsländern bietet der Laptop im Gegensatz zum Tablet die Möglichkeit sich mit einem Computer und der Tastatur vertraut zu machen.

Doch nicht nur im Bezug auf die Hardware fällt meine Entscheidung auf den OLPC. Auch die Software ist besser auf die Bedürfnisse der armen Kinder abgestimmt. Beim OLPC Laptop wird mit selbsterklärenden Symbolen gearbeitet. Dies ist besonders wichtig, da viele der Benutzer weder Lesen noch Schreiben können. Das Tablet gliedert die Applikationen in Traumberufe. In Drittwelt Ländern ist dies nicht angebracht, da viele dieser Berufe gar nicht existieren. Allgemein finde ich persönlich es nicht gut, dass Kinder bereits im jungen Alter zwischen Berufen wählen sollen. Meiner Meinung nach sollte in der Grundausbildung eine breite Basis gelegt werden und die Richtung erst später gewählt werden.

Der XO Laptop ist einfach besser auf die Bedingungen der dritten Welt durchdacht und abgestimmt. Das Tablet richtet sich eher an Kinder in fortgeschrittenen Ländern als an Kinder in Entwicklungsländern.

9 OLPC Einsatzgebiete

Die OLPC Laptops kommen gemäß Angaben von one.laptop.org in folgenden Ländern zum Einsatz.

Tabelle 3 : (Eigene Darstellung) Verteilung OLPC Laptop. Quelle: (One Laptop Per Child)

Land	Anzahl Laptops
Australien	4400
Solomon Islands	300
Vereinigte Staaten von Mikronesien	800
Papua New Guinea	2350
Philippinen	750
Malaysia	100
Sri Lanka	1350
Kambodscha	1000
Thailand	500

Indien	1000
Nepal	6000
China	1000
Mongolei	14500
Pakistan	500
Afghanistan	5000
Iran	343
Bergkarabach	5000
Irak	9150
Libanon	450
Westjordan	4000
Gaza	6000
Italien	600
Äthiopien	6000
Uganda	300
Kenia	500
Rwanda	110000
Mosambik	200
Süd-Afrika	500
Kamerun	1600
Nigeria	6100
Ghana	1000
Mali	300
Argentinien	60000
Uruguay	510000
Paraguay	4000
Brasilien	2600
Peru	860000
Kolumbien	22300
Costa Rica	1500
Guatemala	3000

Nikaragua	25000
Haiti	15000
Mexiko	53700
USA	95100
Total	1843793

Untenstehende eingefärbte Weltkarte bietet einen schnellen Überblick über die Verteilung der OLPC Laptops weltweit.

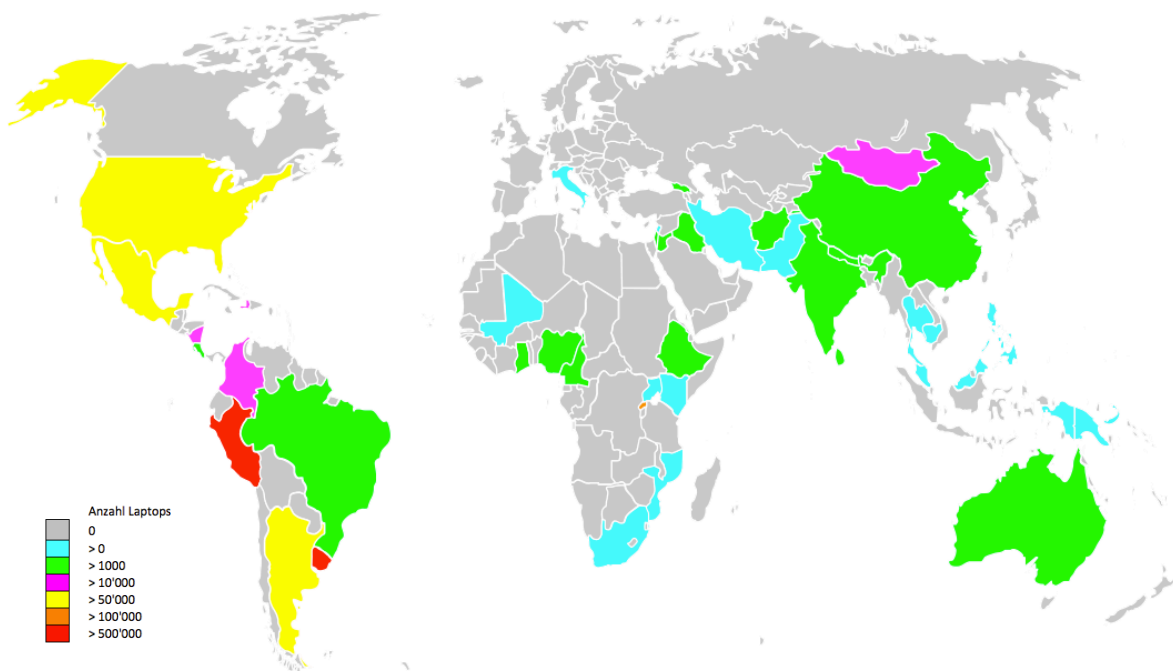


Abbildung 1 : (Eigene Darstellung) Anzahl OLPC Laptops auf der Weltkarte eingefärbt. Quelle: (One Laptop Per Child)

Auf der Weltkarte ist klar ersichtlich, dass sich die Großzahl aller Laptops in Amerika, insbesondere in Südamerika befinden. In Peru (860'000) und in Uruguay (510'000) alleine befinden sich mehr als die Hälfte aller weltweit eingesetzter OLPC Laptops. In Afrika sticht Ruanda mit 110'000 eingesetzten Laptops heraus. In Asien bring es nur die Mongolei auf über 10'000 Laptops.

Im Hinblick auf die Applikation muss die Sprache berücksichtigt werden. Dazu habe ich die jeweiligen Landessprachen der oben aufgeführten Länder herausgesucht und diese dann prozentual anhand der Anzahl Laptops dargestellt. Die jeweiligen Landessprachen wurden dabei mit Hilfe der Google Suchfunktion Amtssprache + Land herausgefunden. Untenstehende Grafik zeigt welche Sprachen in den Ländern in denen OLPC Laptops zum Einsatz kommen am meisten gesprochen werden.

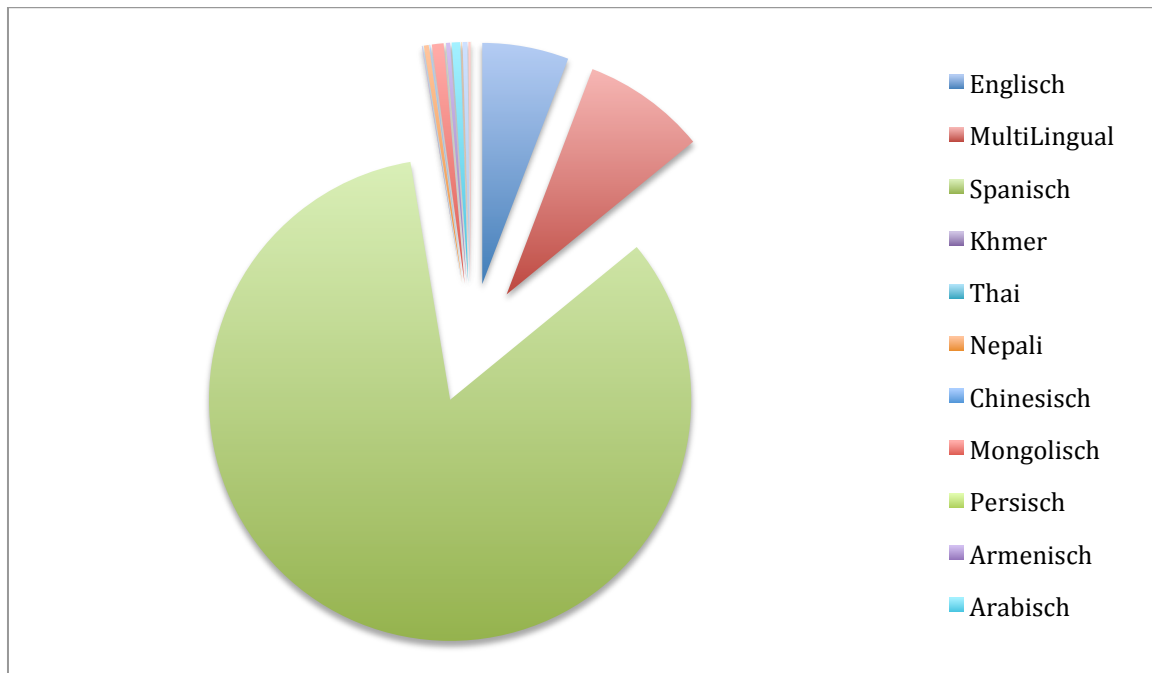


Abbildung 2 : (Eigene Darstellung) Sprachen der Länder in denen OLPC Laptops eingesetzt werden. Quelle: (One Laptop Per Child)

Mehr als 80% der eingesetzten Laptops werden in Ländern eingesetzt in denen Spanisch gesprochen wird. Auffallend ist auch dass in vielen Ländern in denen der OLPC Laptop zum Einsatz kommt mehrere Sprachen gesprochen werden. Dies erschwert es eine Applikation zu entwickeln, die von mehreren Kindern benutzt werden kann. Will man also möglichst vielen Kindern helfen so muss man die Applikation auf jeden Fall in Spanisch anbieten.

Wie die OLPC Laptops in Zukunft verteilt werden ist schwierig vorherzusehen. Grundsätzlich wird eine große Anzahl an Laptops gemäß Verträgen mit der Regierung versendet. Diese Laptops werden anschließend von den Lehrern an die Kinder verteilt. Um herauszufinden wie sich die Verteilung der OLPC Laptops künftig verhalten wird, ist es wichtig aufzuzeigen in welchen Ländern Kinder überhaupt zur Schule gehen können. Denn in Ländern in denen den Kindern der Zugang zur Schule verwehrt bleibt müssen zuerst andere Maßnahmen getroffen

werden. Gemäß Zahlen der UNESCO hatten im Jahre 2011 57 Millionen Kinder keinen Zugang zur Schule. Untenstehende Grafik zeigt wie sich diese 57 Millionen verteilen. (Unesco.org, 2014)

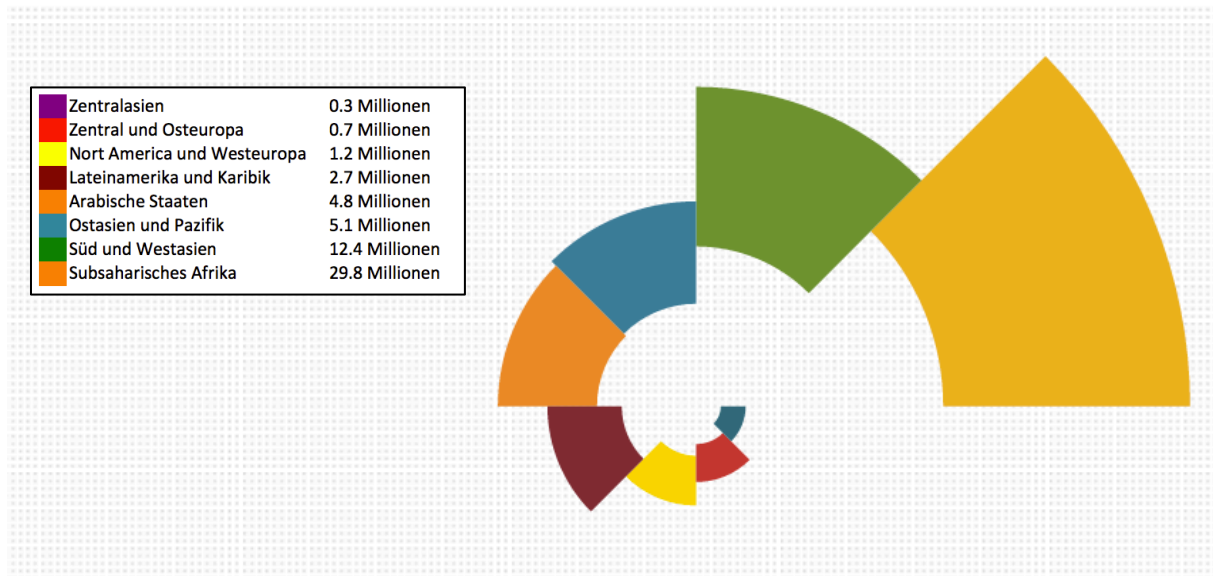


Abbildung 3 : Kinder die nicht zur Schule gehen (Global). Quelle: (Unesco.org, 2014)

10 Zweiter Entscheid – Welche Sprache und welche Länder?

Anhand obiger Ausführungen wird klar dass der Großteil der Kinder welche im Besitz eines OLPC Laptops sind Spanisch sprechen. Um also vielen Kindern zu helfen soll die Applikation vorerst in Spanisch umgesetzt werden.

11 Was wird gebraucht - Welches ist das Bedürfnis?

Dieser Abschnitt bezieht sich auf den EFA Global Monitoring Report (Weltbildungsbericht 2013/14) der UNESCO. Alle Daten die in diesem Kapitel präsentiert werden stammen von folgenden Quellen: (UNESCO, 2014), (unesco.de, 2014). Gemäß diesem Bericht haben immer noch 57 Millionen Kinder keinen Zugang zu Bildung. Nebst dem Zugang stellt ebenso die Bildungsqualität ein großes Problem dar. Ein Drittel aller Grundschüler beherrschen die Grundlegenden Thematiken wie Lesen und einfache Mathematik trotz Schulbesuches nicht. Oft fehlt es an qualifiziertem Lehrpersonal. Den Kindern soll künftig der Zugang zu Bildung nicht verwehrt bleiben. Dazu soll den Kindern in der Grundschule ein Basiswissen mit auf den Weg gegeben werden um künftige Anforderungen im Berufsleben erfolgreich zu meistern oder sich anschließend weiterbilden zu können. (UNESCO, 2014)

Im Jahr 2000 wurden in Dakar am World Education Forum 6 Ziele in Bezug auf die Bildung festgelegt.

- Ziel 1: Ausbau der frühkindlichen Bildung
- Ziel 2: Für alle Kinder soll die Schule obligatorisch sein. Die Kinder sollen eine qualitativ hochwertige Schule besuchen und diese auch abschließen.
- Ziel 3: Umfassende Lernangebote und Angebote zum Training von Basisfähigkeiten sollen die Lernbedürfnisse Jugendlicher und Erwachsener befriedigen.
- Ziel 4: Reduktion der Analphabeten Rate um 50 Prozent. Erwachsene sollen ebenfalls Zugang zu Grund- und Weiterbildungen haben.
- Ziel 5: Gleichberechtigung der Geschlechter im gesamten Bildungsbereich.
- Ziel 6: Verbesserung der Qualität von Bildung.

Quelle: (unesco.de, 2014)

Für jedes der obengenannten Ziele mit Ausnahme des Ziels Nummer sechs wurden Messwerte definiert. Diese Messwerte lauten wie folgt.

- Ziel 1: 80% der Kinder genießen frühkindliche Ausbildung
- Ziel 2: 97% der Kinder genießen eine obligatorische qualitativ hochstehende Grundausbildung.
- Ziel 3: Für 97% der Erwachsenen stehen Weiterbildungsmöglichkeiten zur Verfügung.
- Ziel 4: 97% der Erwachsenen können Lesen
- Ziel 5: Der „Gender parity index“ beläuft sich zwischen 0.97 und 1.03.

Quelle: (UNESCO, 2014)

Gemäß dem Beschluss des Forums sollten alle Ziele bis 2015 erreichen. Trotz vielen Fortschritten wird man kein einziges der gesteckten Ziele bis 2015 erreichen können. Untenstehende Grafik bietet eine Übersicht über die aktuelle Zielerreichung.

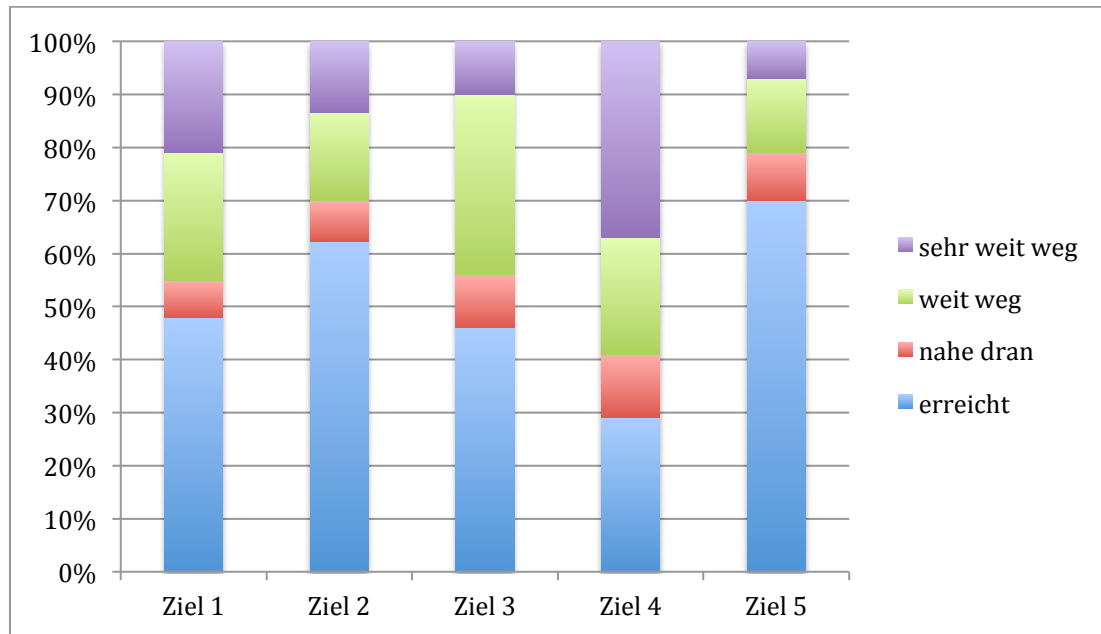


Abbildung 4 : (Eigene Darstellung) Grafik zur Erreichung der EFA Ziele (Stand: 2013/2014) : Quelle (UNESCO, 2014)

Für mich ist besonders das Ziel 2 und Ziel 6 von großer Bedeutung. Diese Arbeit soll mit Helfen die Grundausbildung der Kinder in Bezug auf Qualität und Erreichbarkeit zu verbessern.

Gemäß dem Weltbericht verfügen über 250 Millionen Menschen über kein Basiswissen. Es ist daher von höchster Priorität das sowohl Lesen, Schreiben und Mathematik gefördert werden. Berechnungen des EFA Global Monitoring Teams zeigen, dass wenn alle Kinder der Entwicklungsländer die Grundschule mit grundlegenden Lesefähigkeiten verlassen würden, 171 Millionen Menschen aus der Armut befreit werden könnten. Um der Armut zu entkommen muss das Einkommen steigen. Global gesehen erhöht ein Jahr Schule das Einkommen um 10%. Die Wahrscheinlichkeit, dass ein arbeitender Erwachsener in Armut verfällt, reduziert sich durch eine abgeschlossene Grundausbildung um 20%. In Pakistan verdient eine Frau mit guten Lesefähigkeiten 95% mehr als eine Analphabetin. Doch auch in ländlichen Gegenden ist das Ausmaß der Grundausbildung spürbar. So können Bauern mit grundlegenden Lese- und Mathematikfähigkeiten neue Informationen aufnehmen und diese anschließend in den täglichen Betrieb einbringen. Ein weiterer Vorteil der Bildung ist es, dass die Kinder gebildeter Eltern höchstwahrscheinlich auch eine gute Bildung genießen können. Bildung schützt Bauern davor ausgenommen zu werden. Durch die Fähigkeit zu lesen, können sie faire Verträge abschließen. Gebildete Menschen leben gesünder. Gebildete Menschen sind besser über Krankheiten informiert. (UNESCO, 2014)

Bildung ist also von höchster Wichtigkeit und hilft den Menschen ihre Probleme größtenteils selbst zu lösen. Es gilt besonders die Grundkenntnisse wie Lesen und Mathematik zu fördern. Von den weltweit 650 Millionen Primarschüler haben 250 Millionen nie richtig Lesen, Schreiben und Rechnen gelernt. (UNESCO, 2014)

Untenstehende Grafik zeigt die Prozentuale Darstellung der Anzahl Kinder welche die minimalen Leseanforderungen erfüllt haben. Die Daten aller in diesem Kapitel abgebildeten Grafiken stammen aus dem EFA Global Monitoring Report 2013 / 2014 der UNESCO. Die Daten basieren auf Berechnungen des EFA Global Monitoring Team. Als Grundlage für die Berechnungen dienten die Auswertungen folgender Studien: International Mathematics and Science Study (TIMSS), Progress in International Reading Literacy Study (PIRLS), Southern and Eastern Africa Consortium for Monitoring Educational Quality (SACMEQ), Programme on the Analysis of Education Systems of the CONFEMEN (PASEC), Second Regional Coparative and Explanatory Study (SERCE). Die Unterteilung in einzelnen Grafiken in Arm und Reich beziehen sich dabei auf das jeweils oberste und unterste Quartil in Bezug auf den sozioökonomischen Statusindex der PASEC, SACMEQ und SERCE Studie. Auch die Spezifikation des minimalen Lernstandards bezieht sich auf diese Studien. (UNESCO, 2014)

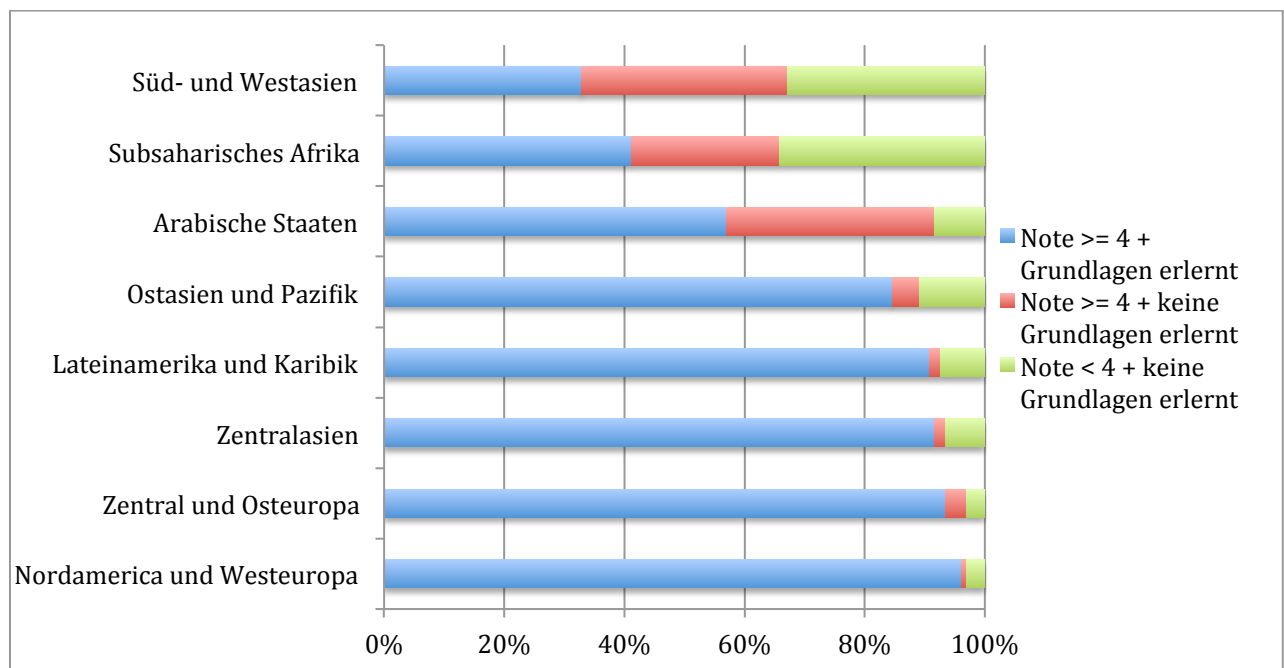


Abbildung 5 : (Eigene Darstellung) Lesefähigkeiten weltweit (Stand: 2013/2014). Quelle: (UNESCO, 2014)

Betrachtet man die obige Grafik so stellt man fest, dass vor allem Süd- und Westasien sowie auch das subsaharische Afrika beim Lese Test sehr schlecht abschneiden. Mehr als die Hälfte der Kinder haben hier nie richtig lesen gelernt. In diesen Ländern sollte man das Augenmerk vor allem auf eine gute Ausbildung in Lesen und Schreiben legen. Leider sind nur aus 85 Ländern alle Daten verfügbar. (UNESCO, 2014)

Um genauere Aussagen machen zu können, habe ich nach einer detaillierteren Statistik gesucht. Eine Statistik die der obigen gleicht, aber in Länder unterteilt ist. Diese Statistik habe ich anschließend mit den Ländern in denen OLPC Laptops im Einsatz sind abgeglichen um auf einen Blick zu sehen in welchem Land Handlungsbedarf in Bezug auf das Lesen herrscht. Leider habe ich nicht zu allen Ländern Daten finden können. Dennoch lassen sich einige interessante Rückschlüsse mittels untenstehender Daten ziehen.

11.1 Subsahara-Afrika

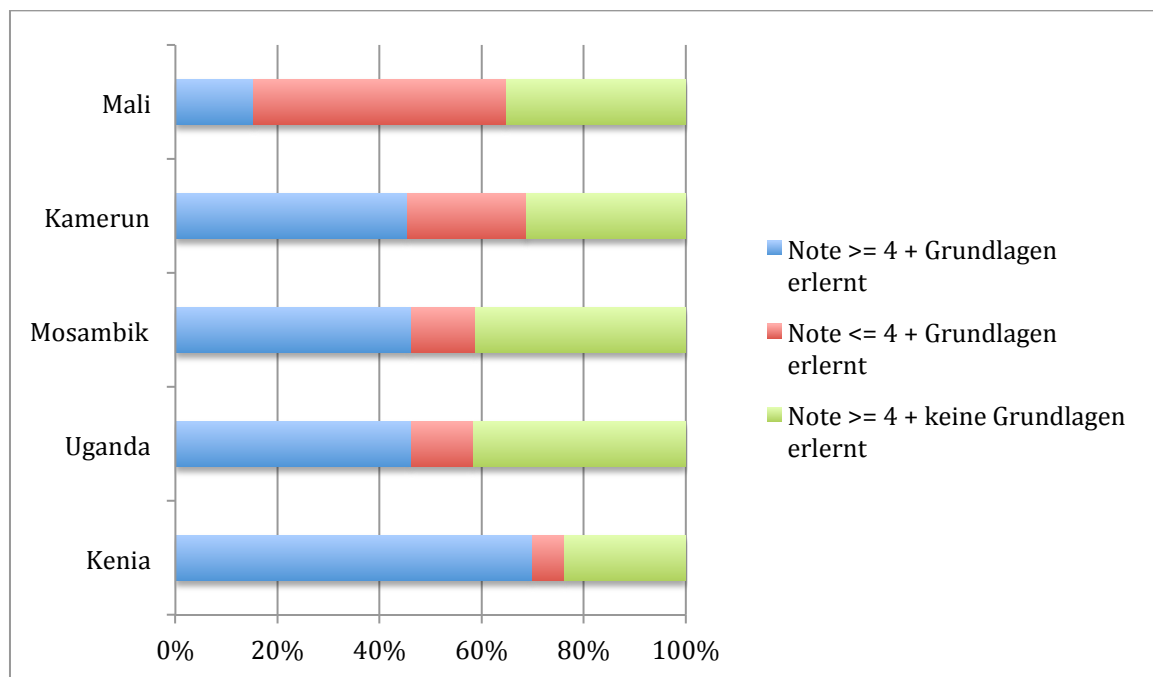


Abbildung 6 : (Eigene Darstellung) Lesefähigkeiten Subsaharisches Afrika. Quelle: (UNESCO, 2014)

In allen Ländern des subsaharischen Afrikas herrscht großer Nachholbedarf im Lesen. Denn mehr als die Hälfte der Kinder lernen die Grundlagen des Lesens nicht. Rund ein Viertel erreichte zwar eine 4, lernte aber trotzdem nie richtig zu lesen. Über ein Drittel erreichte die 4 nicht. Besonders schwierig ist es für Kinder aus ärmlichen Haushalten. Bei Kindern aus wohlhabenden Häusern ist nicht nur die Wahrscheinlichkeit, dass sie die Schule abschließen grösser, sondern auch dass sie in der Schule die Grundlagen lernen. In 15 Ländern erreicht

nur eines der Kinder aus ärmlichen Verhältnissen die letzte Ausbildungsstufe und erlernt dabei die Grundlagen. In Chad beispielsweise schließen nur 2% aller Kinder aus ärmlichen Haushalten die Grundschule ab. Aus wohlhabenden Haushalten hingegen 43%. Durch einen geringeren Grad zwischen Arm und Reich konnte Botswana beispielsweise seine Bildung verbessern. Afrika verfügt über viele verschiedene Sprachen. Kinder welche die Sprache einer Minderheit sprechen sind klar benachteiligt. Viele Kinder werden bereits in den frühen Jahren in einer ungewohnten Sprache unterrichtet. Dies erschwert es den Kindern den Unterrichtsstoff richtig zu verstehen. Um dies zu Umgehen müssen die Lehrmaterialien zweisprachig gestaltet werden. Durch eine schlechte Grundausbildung erhöht sich die Wahrscheinlichkeit, dass die Kinder die Schule früh verlassen. In Äthiopien beispielsweise ist die Wahrscheinlichkeit, dass Kinder welche Mathematik im unteren Viertel abschlossen die Schule verlassen doppelt so hoch. Die Probleme im subsaharischen Afrika liegen auf der Hand. Ein Besuch der Grundschule ist noch lange kein Garant für das Erlernen der Grundlagen. Die Bildungsqualität lässt stark zu Wünschen übrig. Dies wirkt sich besonders in den Mathematik und Lesefähigkeiten aus. So sind im subsaharischen Afrika 40% der jungen Personen nicht in der Lage einen Satz zu lesen. Hingegen hat fast die Hälfte der jungen Leute Ruandas die Schule nicht mehr als vier Jahre besucht. Trotzdem waren mehr als 50% in der Lage zu lesen. Dies zeigt, dass die Qualität der Schulen in Ruanda den anderen voraus ist und Kinder trotz wenig Zeit in der Schule die Grundlagen beherrschen. (UNESCO, 2014)

Um den Personen in Afrika zu helfen wurde die Organisation Hopeville gegründet. Gemeinsam mit dem Verein „Amagu“ aus Gampel helfen sie in Nigeria. Nebst verschiedenen Hilfeleistungen unterrichten Lehrer dieses Vereins ebenfalls vor Ort. Gemäß Aussage von Martin Zeiter einem Vorstehenden Mitglied des Vereins, stehen besonders Englischunterricht und Mathematikunterricht im Fokus. Obwohl Englisch die Landessprache Nigerias ist, wird diese besonders von den jungen Schülern nicht beherrscht. Der Großteil der Schüler spricht „Igbo“, eine lokale Sprache. In Bezug auf Mathematik fehlen ebenfalls jegliche Basiskenntnisse. Durch Mathematikprogramme könnte man aber mehreren Kindern helfen, da man über das gleiche Zahlensystem verfügt. Untenstehende Darstellung zeigt auf wie groß der Nachholbedarf in Mathematik tatsächlich ist. Die Grafik zeigt wie viele Kinder jeweils die Grundlagen der Mathematik beherrschen. Die Daten stammen vom EFA Global Monitoring Report. Dabei habe ich die Grafik so angepasst, dass nur Länder abgebildet werden in denen auch OLPC Laptops zur Verfügung stehen.

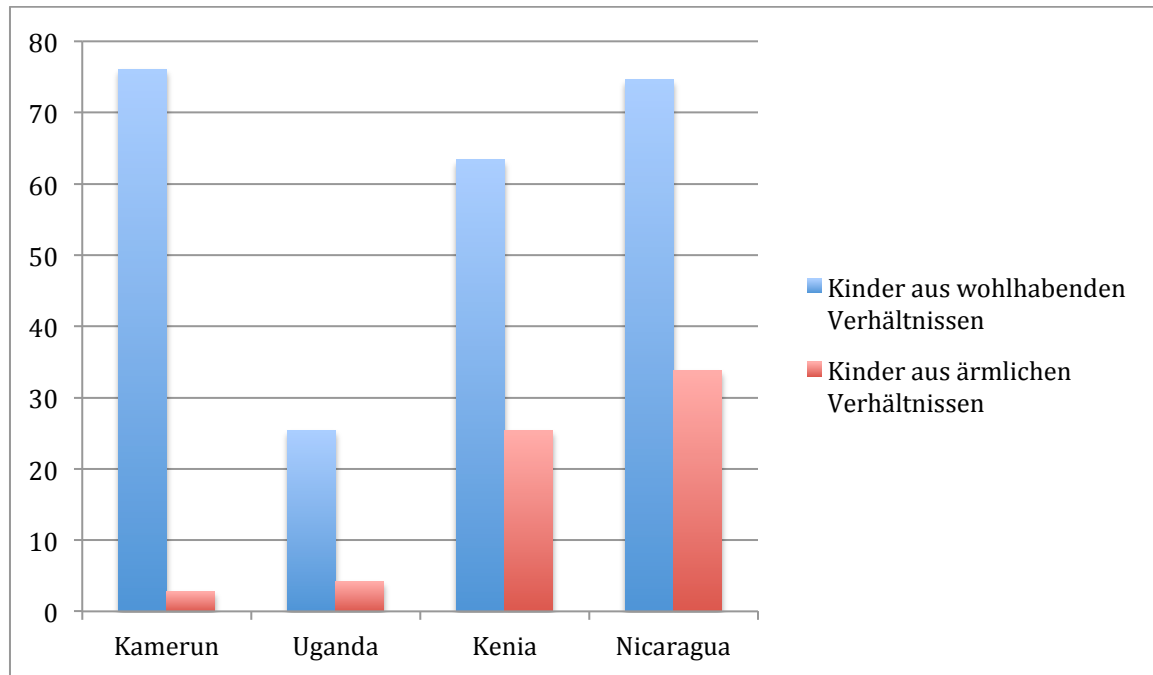


Abbildung 7 : (Eigene Darstellung) Subsahra-Afrika / Anzahl Kinder welche die Grundlagen der Mathematik beherrschen. Quelle: (UNESCO, 2014)

Fast 30 Millionen Kinder haben keinen Zugang zur Schule. Da die OLPC vorwiegend über die Schule verteilt werden, ist es zudem schwierig, dass die Kinder überhaupt in den Besitz eines OLPC Laptops kommen. Dies zeigt sich auch in der Anzahl OLPC Laptops in afrikanischen Ländern. Einzig und allein in Ruanda sind mehr als 10'000 Laptops verteilt. Nigeria (6'100), Äthiopien (6'000) und Ghana (1'000) sind nach Ruanda die einzigen Länder die mehr als oder genau 1000 Laptops haben. In Afrika gilt es vor allem den Zugang zu Bildung zu verbessern. OLPC Laptops können dann anschließend eingeführt werden. (UNESCO, 2014)

11.2 Lateinamerika und Karibik

Im Gegensatz zu den afrikanischen Ländern ist in Südamerika der Zugang zur Bildung sehr gut. 1.2 Millionen Kinder haben noch keinen Zugang zur Schule. Das bedeutet, dass nur in Zentralasien, Nordamerika und Europa mehr Kinder zur Schule gehen. Latein- und Südamerika sowie die Karibik ist für meine Arbeit besonders interessant, da sich hier die Großzahl aller OLPC Laptops befindet. (UNESCO, 2014)

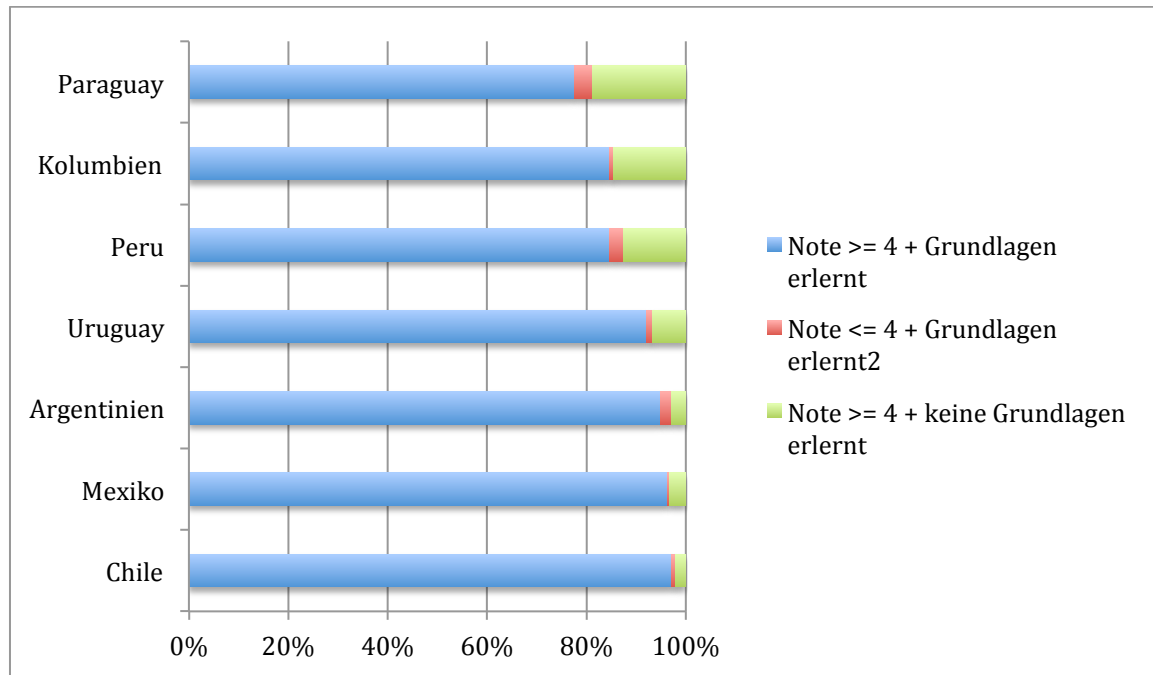


Abbildung 8 : (Eigene Darstellung) Lesefähigkeit Südamerika. Quelle: (UNESCO, 2014)

Die Lesefähigkeiten in Lateinamerika und der Karibik sind sehr ausgeprägt. 91% erreichten eine 4 und erlernten dazu noch die Grundlagen. 1.5% erreichte eine 4 aber erlernte die Grundlagen nicht. 7.5% Erreichten sowohl die 4 nicht und erlernten auch die Grundlagen nicht. Obenstehende Grafik zeigt die Unterschiede der Länder klar auf. Im Vergleich der Lesefähigkeiten fällt besonders auf das Paraguay, Kolumbien und Peru den anderen Ländern hinterherhinken. (UNESCO, 2014)

Betreffend Grundlagen im Lesen ist man auf einem guten Weg. Doch die Performance in Mathematik sieht da schlechter aus. Nur 70 von 100 Kindern erlernen die Grundlagen der Mathematik. So können in Nicaragua 60% der Kinder lesen, aber nur 37% rechnen. Diese Tatsache widerspiegelt sich auch in Chile. Hier haben fast alle Kinder in der Grundschule lesen gelernt. Doch auch hier beherrschen immer noch 20% aller Kinder die Grundlagen der Mathematik nicht. (UNESCO, 2014)

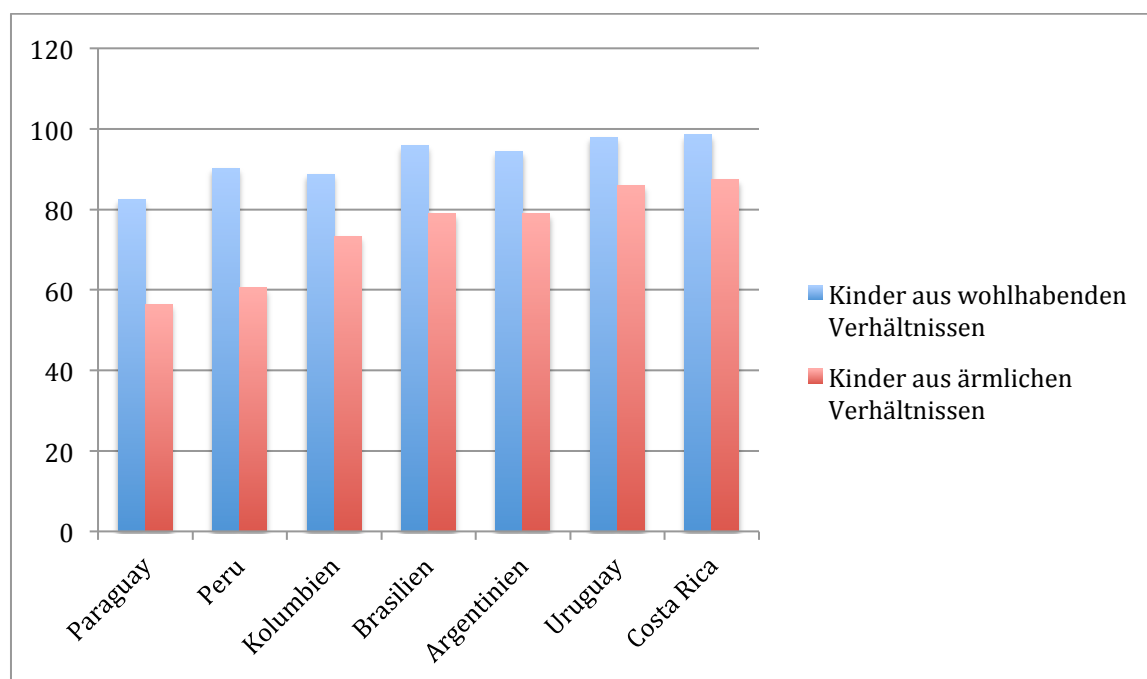


Abbildung 9 : (Eigene Darstellung) Lateinamerika und Karibik / Anzahl Kinder welche die Grundlagen der Mathematik beherrschen. Quelle: (UNESCO, 2014)

Die mathematischen Fähigkeiten sind sehr wichtig. Schüler welche in Mathematik im unteren Viertel abschnitten verließen die Schule viel früher als andere. Auch hier macht die Herkunft einen großen Unterschied. In El Salvador zum Beispiel schließen 42% der Kinder aus armen Verhältnissen die Grundschule ab und erlernen die Grundlagen. Dasselbe gilt für 84% der Kinder aus wohlhabenden Häusern. (UNESCO, 2014)

Peru, Kolumbien und Paraguay weisen sowohl bei den Grundlagen der Mathematik als auch bei den Grundlagen des Lesens die schlechtesten Testergebnisse auf. Die Hauptmankos der Kinder dieser Länder liegen klar beim Lesen und Schreiben sowie in der Mathematik. (UNESCO, 2014)

Gemäß einem Interview mit Otto Imsand, der 17 Jahre lang in Peru unterrichtete, gibt es allerdings auch in der Allgemeinbildung große Lücken. Die Allgemeinbildung ist besonders für Schüler welche später eine weiterführende Schule besuchen möchten von großer Bedeutung. Um in eine solche weiterführende Schule aufgenommen zu werden ist ein gutes Allgemeinwissen Voraussetzung. Der Fokus muss ganz klar in der Verbesserung der Basiskenntnisse in Mathematik und Schreiben liegen, doch auch das Allgemeinwissen ist ein wichtiger Bestandteil der Grundausbildung. (Imsand, 2014), (UNESCO, 2014)

In den drei Ländern mit den größten Defiziten (Peru, Kolumbien und Paraguay) alleine sind derzeit 886'300 Laptops im Einsatz. Darüber hinaus wird in allen Ländern Spanisch gesprochen. Mit einer Lern- App in Spanisch auf einem OLPC Laptop kann man hier sehr vielen Kindern eine Hilfestellung bieten. (UNESCO, 2014)

11.3 Süd- West Asien

In Süd und Westasien 33% der Kinder im Grundschulalter erreichten eine 4 und erlernten die Grundkenntnisse des Lesens. 34% erreichte zwar eine 4, erlernte aber nie die Grundlagen des Lesens. Die restlichen 33% erreichten weder die Stufe 4 noch erlernten sie die Grundlagen des Lesens. Auf die Länder in denen OLPC Laptops vorhanden sind teilt sich dies wie folgt auf. (UNESCO, 2014)

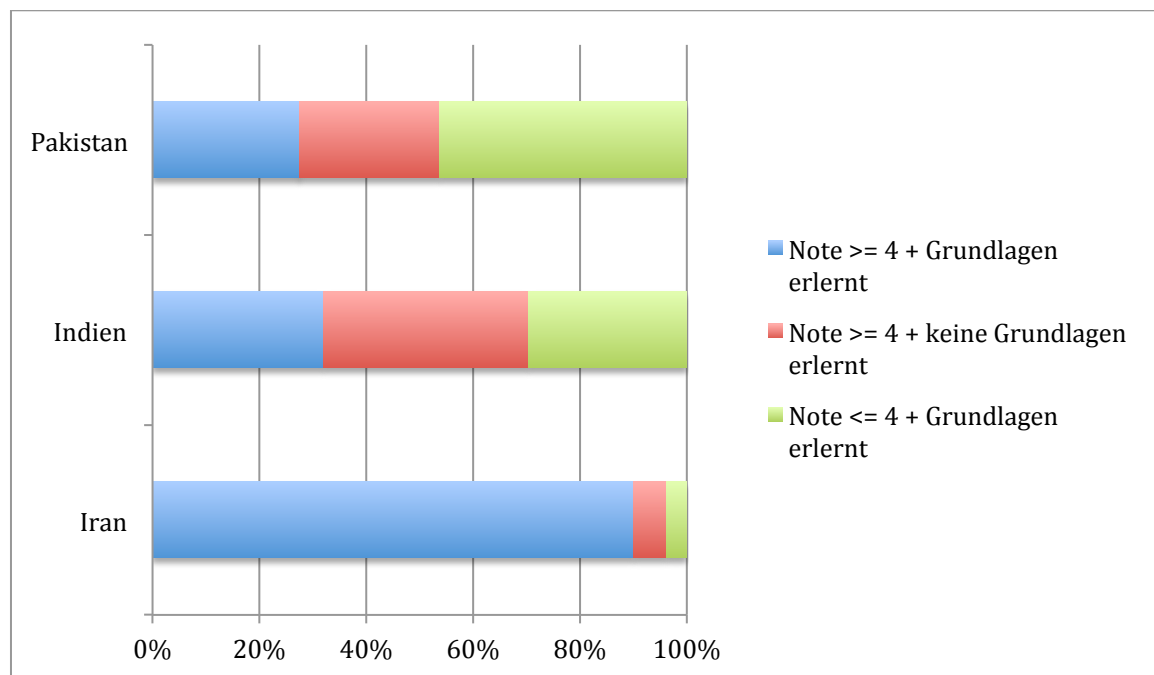


Abbildung 10 : (Eigene Darstellung) Lesefähigkeit Süd-West Asien. Quelle: (UNESCO, 2014)

In Süd West Asien herrscht vor allem in Indien und Pakistan Handlungsbedarf. In Pakistan und Indien hat die Hälfte aller Kinder nie richtig lesen gelernt. Iran schneiden im Gegensatz zu Indien und Pakistan sehr gut ab. Gemäß dem EFA Global Monitoring Report sind die Mathematikgrundlagen noch schlechter als die Lesegrundlagen. Leider stellt der Bericht dazu keine Zahlen zur Verfügung. Auch hier ist der Zugang zu Bildung bereits problematisch. 12.4 Millionen Kinder können hier nicht zur Schule gehen. (UNESCO, 2014)

11.4 Zentralasien

In Zentralasien haben nur 0.3 Millionen keinen Zugang zur Schule. Damit liegt Zentralasien noch vor Nordamerika und Europa. Doch nicht nur der Zugang zur Schule ist gut auch die Bildungsqualität ist hoch. Dies spiegelt sich in den Zahlen der Grundlegenden Kenntnisse des Lesens wieder. (UNESCO, 2014)

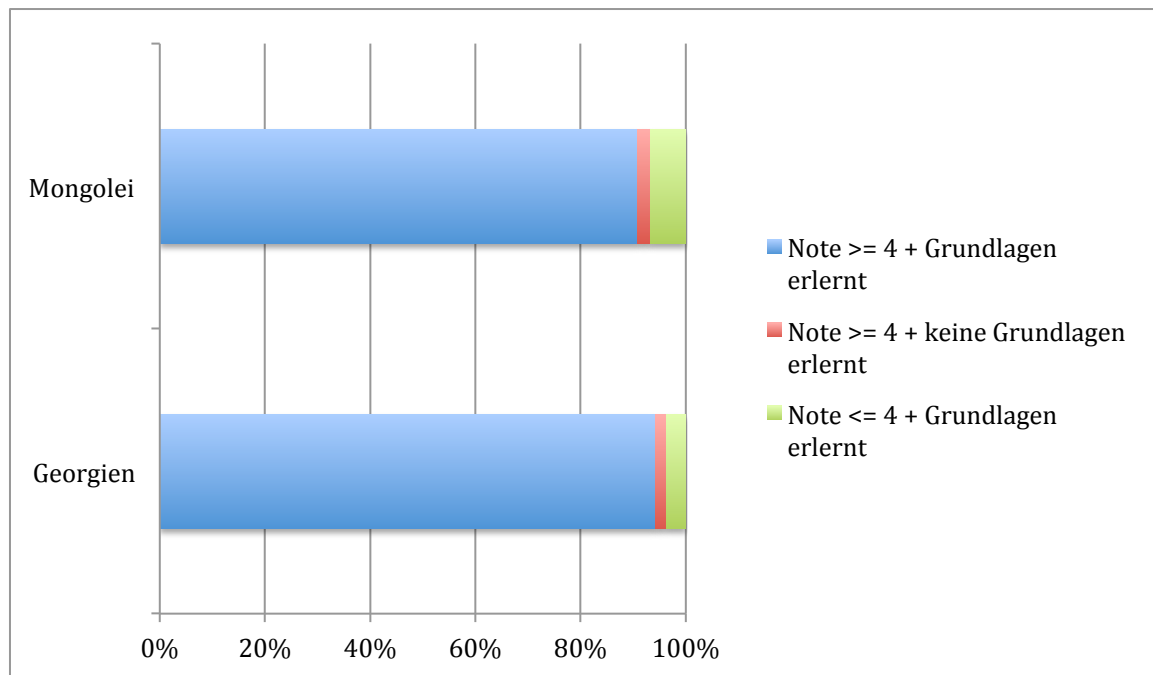


Abbildung 11 : (Eigene Darstellung) Lesefähigkeit Zentralasien. Quelle: (UNESCO, 2014)

Die Testresultate in Zentralasien weisen im Vergleich zu Süd-Westasien, Südamerika und dem subsaharischen Afrika sehr gute Resultate auf.

12 Dritter Entscheid – Welches Bedürfnis soll gedeckt werden

Im vorherigen Abschnitt wurde aufgezeigt welche Probleme in welchen Ländern herrschen. Grundsätzlich kann man folgende Rückschlüsse ziehen. In den afrikanischen, insbesondere den Ländern des subsaharischen Afrikas herrscht großer Handlungsbedarf. In 17 dieser Länder beherrschen weniger als die Hälfte der Kinder die Grundlagen im Lesen, Schreiben und in der Mathematik. Der Handlungsbedarf ist besonders in diesen Ländern groß. Weitere Länder bei denen ebenfalls weniger als die Hälfte die Grundlagen beherrschen sind Indien, Mauritien, Marokko und Pakistan. Besonders die geringe Verbreitung von OLPC Laptops und der generell schwierige Zugang zur Bildung ist in diesen Ländern problematisch.

Unterschiedlichste Sprachen erschweren die Implementation einer Applikation für diese Länder.

Ganz anders ist die Lage in Südamerika. Hier befindet sich der Großteil aller verteilter OLPC Laptops. Zudem wird in praktisch allen Ländern vorwiegend Spanisch gesprochen. Doch in Südamerika sehen die Bedürfnisse etwas anders aus als in Afrika. In Südamerika hat der Großteil der Kinder die Grundlagen des Lesens erlernt. So haben sowohl in Uruguay, Argentinien, Mexiko und Chile mehr als 90% die Grundlagen erlernt. Einzig Peru, Kolumbien und Paraguay hinken hinterher. In Südamerika herrscht eher der Nachholbedarf im Bereich der Mathematik. Nur 70 von 100 Kindern lernen hier die Grundlagen. Weiter fehlt es oft an Allgemeinbildung um später eine weiterführende Schule zu besuchen.

Aufgrund dieser Auswertungen wird klar, dass besonders in Afrika Hilfe nötig ist. Hier aber praktisch keine OLPC Geräte verbreitet sind und daher zuerst andere Maßnahmen getroffen werden müssen. Einzig in Ruanda gibt es viele OLPC Laptops. Das Problem von Ruanda ist allerdings, dass das Land über drei Sprachen verfügt, Kinyarwanda, Französisch und Englisch. Aus diesen Gründen entscheide ich mich dafür eine Mathematikapplikation für südamerikanische Länder zu implementieren. Dadurch kann der größten Anzahl an Kindern geholfen werden.

Mathematik spielt in unserem täglichen Leben eine fundamentale Rolle und ist deshalb von höchster Wichtigkeit. Mathematik bildet die Grundlage für sämtliche Geld- und Geschäftsangelegenheiten. Jemand der die Grundlagen der Mathematik nicht beherrscht, kann keine Finanzplanung vornehmen. Zahlenkenntnisse sind von wichtigster Bedeutung da wir ihnen täglich begegnen. Auch in Entwicklungsländern begegnet man Mathematik täglich. Durch Mathematikgrundlagen hat ein Bauer die Möglichkeit Tatbestände beispielsweise bezüglich Nutzung von Flächen zu berechnen und diese zu optimieren. Doch nicht nur das Zahlenverständnis ist grundlegend sondern auch Geometrie. Mathematik ist für den Mensch der heutigen Zeit unverzichtbar.

13 Technologien

13.1 Phyton

Phyton ist eine höhere Programmiersprache. Phyton unterstützt sowohl objektorientierte, Aspekt orientierte als auch funktionale Programmierung. Phyton wurde Anfangs der 90er Jahre entwickelt. Im Vergleich zu anderen Programmiersprachen wie Java oder auch PHP erscheint die Programmstruktur ohne Klammern. Phyton wurde mit dem Ziel der Einfachheit und Übersichtlichkeit konzipiert. Dies soll durch wenige Schlüsselwörter und auf Übersichtlichkeit optimierte Syntax erreicht werden. Daher wird Phyton oft auch als eine gute Programmiersprache für Anfänger gehandelt. (www.wikipedia.de, 2014)

Der Großteil der Sugar Umgebung wurde in der OLPC Phyton Umgebung geschrieben. Zum großen Vorteil für die Entwickler werden Phyton Bibliotheken angeboten welche das Sugar Application Programming Interface (API) kapseln. (laptop.org, 2014)

Sugar Aktivitäten können in Phyton programmiert werden. Die Benutzeroberfläche der Aktivität kann mit Hilfe des Benutzerschnittstellen-Designers „Glade“ erstellt werden. Glade ist ein Rapid Application Development (RAD) Tool das eine schnelle und einfache Entwicklung grafischer Benutzeroberflächen für die Gimp Toolkit + (GTK+) und Gnome Desktop Umgebung bietet. Die in Glade erstellten Benutzeroberflächen werden als XML gespeichert. Mit Hilfe des GTKBuilder können GTK Objekte erstellt werden die zur Laufzeit dynamisch geladen werden können. (glade.gnome.org, 2014)

13.2 Sugar Web

Bei Sugar Web werden die gängigen Technologien wie Hypertext Markup Language (HTML), JavaScript und Cascading Style Sheets (CSS) eingesetzt. HTML ist die Markup Sprache fürs Web. Mit Hilfe von HTML werden die verschiedenen Elemente auf einer Webseite definiert. Mit CSS wird anschließend das Design dieser Elemente festgelegt. JavaScript bringt die Dynamik in die Web Aktivität. Durch JavaScript können Elemente dynamisch angezeigt werden. (Developer Sugarlabs)

Gemäß Christoph Guéret ist Sugar Web auf dem aktuellen OLPC noch nicht verfügbar. Gemäß Diskussionen des Sugar Camp Paris vom 12. Und 13. April 2014 soll künftig auch

Sugar Web auf einem OLPC XO Laptop verfügbar sein. Sugar soll künftig nicht mehr einfach die Gtk + Python + Fedora Paarung sein. Es soll vermehrt stark auf Sugar Web gesetzt werden. Sugarized bringt neue Perspektiven. Diese ermöglichen es Sugar Web Aktivitäten auf jedem Gerät mit einem HTML 5 Browser zu starten. Darüber hinaus gibt es noch die Web basierenden Betriebssysteme wie WebOS oder Boot2Gecko. Diese bilden eine Alternative zu Fedora + X. Da der Großteil der Verteilten Laptops noch auf XO-1 läuft, könnte dieses, auf Webtechnologie basierende Betriebssystem, XO-1 am Leben erhalten. (The World Wide Semantic Web)

Eine Sugar Web Aktivität ist eine Aktivität die Mithilfe der Standard Webtechnologien wie HTML, JavaScript und CSS erstellt wurde. Diese Aktivitäten werden aufgrund der eingesetzten Technologie als „Web“ Aktivitäten bezeichnet. Sie funktionieren aber auch offline. Diese Aktivitäten werden in der Sugar Shell ausgeführt. Die Sugar Shell ist eine Aktivität welche zur Laufzeit ausgeführt wird. Sie basiert auf den Standard Browser Technologien.

Hauptaufgabe ist das Rendering des Bildschirms, das assistieren bei Events und die Interpretation von JavaScript.

(Developer Sugarlabs)

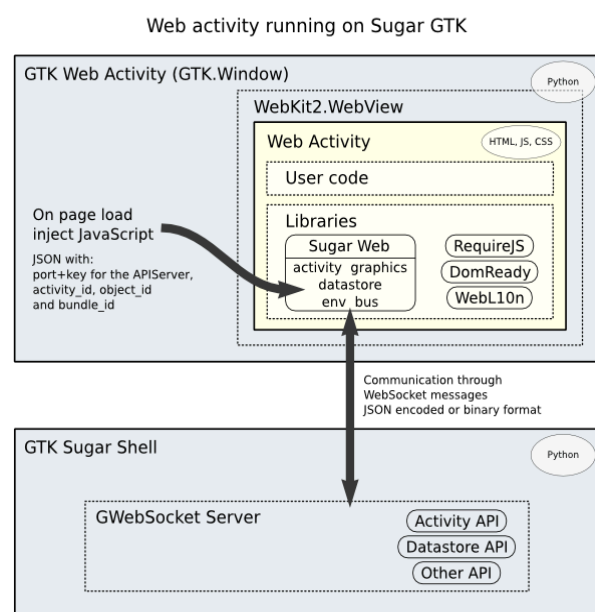


Abbildung 12 : Sugar Web Architektur. Quelle: (Developer Sugarlabs)

14 Vierter Entscheid – Welche Technologie soll benutzt werden?

Bisher wurden Sugar Aktivitäten für den OLPC Laptop mit Python und Fedora implementiert. Infolge diverser Diskussionen am Sugar Camp in Paris will OLPC künftig auf Sugar Web setzen. Ein Vorteil ist, dass die Aktivität auf jedem Gerät mit HTML 5 fähigen Browser ausführbar ist. Dies bedeutet dass es nicht nur auf dem OLPC Laptop oder dem OLPC Tablet ausführbar ist, sondern die Aktivität kann auch auf jedem normalen Computer mit Sugarizer ausgeführt werden. Sugar will in Zukunft vermehrt auf Webtechnologien setzen. Dies wurde

mir auch von Christoph Guéret, einem Forscher der Data Archiving and Network Services (DANS) aus Holland welcher stark im OLPC Projekt involviert ist in einer Mail vom 23. April 2014 bestätigt. Gemäß Christoph Guéret ist es aktuell ein sehr guter Zeitpunkt um auf Sugar Web umzusteigen. Wie bereits erwähnt können Sugar Web Aktivitäten derzeit noch nicht auf dem XO Laptop gestartet werden. Jedoch kann man bereits Sugar Web Aktivitäten implementieren und diese mit Sugarizer testen. (The World Wide Semantic Web)

Basierend auf obigen Informationen beschlossen Fabio Cretton, Laurent Sciboz und ich an der Sitzung vom 28.04.2014, dass meine Anwendung in Sugar Web implementiert wird.

15 Fünfter Entscheid – Welche Entwicklungsumgebung wird gebraucht?

Um eine neue Sugar Web Aktivität zu implementieren stehen zwei mögliche Entwicklungsumgebungen zur Verfügung. Diese sind osbuild und Sugarizer. Osbuild ist die „Sugar Build Environment“. Osbuild stellt eine volle Sugar Umgebung zur Verfügung. Osbuild sollte dann gewählt werden, wenn man eine vollständige Entwicklungsumgebung auf einem GNU Linux haben möchte. Sugarizer hingegen simuliert die Sugar Umgebung in einem herkömmlichen Browser. Um mit Sugarizer zu entwickeln braucht man also nur einen Browser. Der Nachteil von Sugarizer ist es, dass man nicht direkt in einer Sugar Umgebung arbeitet. Ein weiterer Vorteil von osbuild ist es, dass nach Abschluss des Projekts ein Bundle generiert werden kann und dieses anschließend auf den „Sugar Activities Market“ hochgeladen werden kann. (Sugarlabs)

Aufgrund dieser Vorteile verwende ich in meiner Arbeit osbuild um eine voll Sugar Umgebung zu simulieren.

16 Die neue Applikation

16.1 State of the art - Liste der Webaktivitäten

Einzigartigkeit ist ein besonders wichtiges Kriterium bei der Implementierung einer neuen Sugar Web Aktivität. Um sicherzustellen, dass die Aktivität die ich entwickeln möchte in Sugar bisher nicht existiert habe ich alle aktuellen Sugar Aktivitäten aufgelistet. Gemäß Sugarlabs gibt es derzeit 17 Aktivitäten für Sugar Web (Stand 29. April 2014). Alle in diesem Kapitel stehenden Informationen stammen von folgender Quelle: (Sugarlabs.org)

Welcome Web 2

Welcome Web 2 ist eine Aktivität die eine erste Hilfestellung für Sugar bietet. (Sugarlabs.org)

Install Sugar Web Services 1

Diese Aktivität erlaubt die einfache Installation von Sugar Web Services ohne den Gebrauch eines Terminals. (Sugarlabs.org)

Gravatar 4

Diese Aktivität soll zeigen wie einfach es ist einen Avatar von „Gravatar“ zu bekommen. (Sugarlabs.org)

Tank Operation 1

Tank Operation ist ein Arkade Spiel das den Kindern bei der Verbesserung ihrer mathematischen Fähigkeiten helfen soll. Das Spiel basiert auf der Idee von Tuxmath von Tux4Kids. Das Spiel beinhaltet verschiedenste Missionen. Jede dieser Missionen erlaubt es dem Kind eine Mathematikoperation zu trainieren. Anschließend werden diverse Mathematikaufgaben gestellt. Wird die Frage richtig beantwortet, so zerschießt der Panzer das Ziel mit dem angegebenen Resultat. Ziel des Spiels ist es, spielerisch einen Lerneffekt zu erzielen. (Sugarlabs.org)

2048

2048 sorgt seit kurzer Zeit für Begeisterung. Das Originalspiel wurde vom 20 jährigen Gabriele Cirulli entwickelt und ist für alle Geräte verfügbar. 2048 für Sugar braucht dieselbe

Spiel-Engine wie das Original beinhaltet zusätzlich aber kleine Änderungen für Sugar. Ziel dieses Spiels ist es Zweierpotenzen durch geschicktes Hin und Her schieben von Zahlenquadraten immer weiter zu verdoppeln. (Sugarlabs.org)

Slides 2

Slides ermöglicht die Erstellung und Bearbeitung von Präsentationen in Sugar. Slides überzeugt durch simple Bearbeitungstools, Vorhandene Foliendesigns für Präsentationen und eingebundene Bilder mit Bildunterschriften. Erstellte Präsentationen können einfach per Knopfdruck in der Cloud gespeichert und so mit anderen Teilnehmern geteilt werden. (Sugarlabs.org)

Food Chain 4

Food Chain ist ein Spiel zur Erlernung von Tieren und deren Nahrungskette (Wer frisst was?). Die Aktivität besteht aus 3 kleinen Spielen mit 20 Levels welche eine Komplexitätssteigerung beinhalten. (Sugarlabs.org)

Ziel des ersten Spiels ist es Tiere anhand ihrer Nahrung einzuordnen. So wird ein Tiger beispielsweise bei den Fleischfressern eingeordnet. Beim zweiten Spiels geht es darum Tiere nach der Fragestellung „Wer frisst was?“ einzuordnen. Z.B Schlange frisst Frosch, Frosch frisst Fliege. Beim dritten Spiel handelt es sich um ein Arkade Spiel. Hier muss man beispielsweise als Frosch allen Schlangen ausweichen und dabei möglichst viele Fliegen fressen. (Sugarlabs.org)

Learn to Code

Mit Hilfe dieser Aktivität kann man lernen seine eigenen Aktivitäten mit JavaScript zu schreiben. Nebst einer Einführung über JavaScript generell werden zudem konkrete Beispiele gezeigt. (Sugarlabs.org)

Grid Paint 2

Grid Paint 2 ist eine einfach gestaltete Aktivität die Kindern das Zeichnen ermöglicht. Dabei werden allerdings keine Freihandzeichnungen erstellt. Die Zeichnungen basieren auf einem aus Dreiecken bestehenden Gitter. Das Kind kann aus dem Seitenmenü eine Farbe auswählen und anschließend eines der Dreiecke in der gewünschten Farbe einfärben. (Sugarlabs.org)

Clock Web 1

Die Aktivität Clock Web 1 zeigt die Zeit in Form einer analogen Uhr an. (Sugarlabs.org)

Memorize Web 1

Mit der Aktivität Memorize Web 1 können die Mathematikfähigkeiten verbessert werden. Die Aktivität beruht auf der Idee von Memory. So muss beispielsweise $2+4$ zusammen mit 6 aufgedeckt werden. (Sugarlabs.org)

Stopwatch Web 1

Die Stopwatch Web 1 Aktivität ist, wie der Name bereits verrät, eine Stoppuhr. Diese Stoppuhr kann mehrere Zeiten wie beispielsweise Rundenzeiten speichern. (Sugarlabs.org)

Paint Web

Die Paint Web Aktivität ermöglicht es Freihandzeichnungen herzustellen. (Sugarlabs.org)

Gears 6

Ziel von Gears 6 ist es Zahnräder mit Ketten so zu verbinden, dass sich alle Zahnräder drehen. Mit dieser Aktivität soll das logische Denkvermögen gefördert werden. (Sugarlabs.org)

Markdown Activity 3

Mit Hilfe dieser Aktivität wird in Markdown erfasster Text automatisch in HTML übersetzt. (Sugarlabs.org)

Install Activities 3

Mit Hilfe dieser Aktivität können die verschiedenen Sugar Web Aktivitäten installiert werden. (Sugarlabs.org)

LOL (Last One Loses)

Last One Loses wird gegen den Computer gespielt. Zu Beginn werden eine zufallsbestimmte Anzahl an Hölzern angezeigt. Anschließend zieht jeder Spieler abwechselungsweise 1-3 Hölzer. Derjenige der zuletzt noch ein Holz übrig hat verliert. (Sugarlabs.org)

An der obenstehenden Liste ist zu erkennen, dass es noch einen großen Spielraum für neue Sugar Web Aktivitäten gibt.

16.2 Was wird im Bereich der Mathematik gebraucht?

16.2.1 Massachusetts Curriculum Framework for Mathematics

Damit die Applikation auch wirklich in der Praxis gebraucht wird, ist es wichtig die Aktivitäten auf Lernziele anzupassen. Rahmenlernpläne definieren die Lernziele sowie die Lerninhalte. Im Bereich der Mathematik verwendet SugarLabs den „Massachusetts Curriculum Framework for Mathematics. Als Maßstab wird Stufe 4 genommen. Die Stufe 4 beinhaltet folgende Themen und stellt folgende Anforderungen. Quelle: (Wiki Sugarlabs, 2014)

Untenstehende Angaben stammen alle von folgender Quelle: (Massachusetts Curriculum Framework for Mathematics, 2011)

16.2.2 Operationen und algebraisches Denken

Die 4 Grundoperationen (Addition, Subtraktion, Multiplikation und Division) müssen so eingesetzt werden können um konkrete Fragestellungen und Probleme zu beantworten und zu lösen.

- *Eine Gleichung als Vergleich interpretieren können. Z.B $35=5 * 7$. Dem Lernenden muss bewusst sein, dass sowohl 5 mal eine 7, 35 ergibt als auch 7 mal eine 5.*
- *Gleichungen mit Hilfe der vier Grundoperationen sollen gelöst werden können. Aufgaben bei denen die einzusetzende Zahl durch ein x repräsentiert. Z.B $5 * x = 25$ müssen gelöst werden können.*
- *Textaufgaben zu den vier Grundoperationen sollen beantwortet werden können.*

Mit Faktoren und Vielfachen muss man vertraut sein.

- *Alle Faktorpaare im Bereich von 1-100 müssen gefunden werden können. Dem Lernenden muss klar sein, dass eine Ganzzahl ein Produkt seiner Faktoren ist. Er muss herausfinden können ob eine Zahl wie Z.B 60 ein Vielfaches von 12 ist. Darüber hinaus soll er in der Lage sein, im Bereich von 1-100 die Primzahlen zu bestimmen.*

Muster sollen erstellt und analysiert werden können.

- *Der Schüler soll ein „Muster“, welches auf einer spezifischen Regel basiert erstellen können. Z.B gibt man dem Schüler das „Muster“ füge drei hinzu sowie die Startzahl eins, so muss er in der Lage sein das Muster 1,4,7,10,13 etc... aufzulisten.*

Quelle dieser Angaben: (Massachusetts Curriculum Framework for Mathematics, 2011)

16.2.3 Nummern und Operationen der Zehnerbasis

Generelles Verständnis mehrstelliger Ganzzahlen muss vorhanden sein.

- *Der Schüler soll verstehen, dass bei einer mehrstelligen Zahl eine Ziffer mehr gleichbedeutend mit dem Zehnfachen der Ausgangszahl ist. Z.B $500 : 50 = 10$.*
- *Mehrstellige Zahlen sollen miteinander verglichen werden. Dabei ist der Schüler in der Lage zu bestimmen, ob welche Zahl grösser ist oder ob die Zahlen gleich Groß sind. Um diesen Vergleich darzustellen benutzt der Schüler die Zeichen $>$, $<$ und $=$.*
- *Der Schüler soll in der Lage sein mehrstellige Zahlen richtig einzuordnen und anschließend die richtigen Rundungsoperationen vorzunehmen. Z.B 58 soll auf die nächste durch 10 teilbare Zahl aufgerundet werden.*

Mit mehrstelligen Ganzzahlen muss gerechnet werden können.

- *Der Schüler soll in der Lage sein auf Addition und Subtraktion auf mehrstelligen Zahlen mit Kopfrechnen auszuführen.*
- *Mehrstellige Zahlen bis zu 4 Stellen sollen mit einer einstelligen Zahl, und zweistellige Zahlen mit einer zweistelligen Zahl multipliziert werden können. Die Rechnungen sollen mit Tabellen verdeutlicht werden können. Der Schüler soll so in der Lage sein mit einer 12×12 Tabelle zu rechnen.*
- *Mehrstellige Zahlen bis zu 4 Stellen sollen durch eine einstellige Zahl dividiert werden. Dabei soll der Zusammenhang zwischen Multiplikation und Division verstanden werden.*

Quelle dieser Angaben: (Massachusetts Curriculum Framework for Mathematics, 2011)

16.2.4 Bruchrechnen

Über generelles Verständnis des Bruchrechnens und der Gleichwertigkeit verfügen.

- *Der Schüler soll in der Lage sein zu erklären warum $4/5$ gleichbedeutend mit $2 \cdot 2/2 + 3$ ist.*
- *Brüche sollen mit Hilfe der Zeichen $<$, $>$ und $=$ verglichen werden.*

Brüche aus einheitlichen Brüchen mit Hilfe von Operationen berechnen können.

- *Verstehen wie man Brüche addiert und subtrahiert. Z.B 2 und $1/8 = 1 + 1 + 1/8$.*
- *Basismultiplikationen auf Brüchen ausführen. Z.B $5 \cdot 1/4 = 5/4$.*
- *Der Schüler muss in der Lage sein Textaufgaben zu diesem Thema zu beantworten.*

Dezimalnotation für Brüche beherrschen und Zusammenhänge zwischen Brüchen und Dezimalzahlen verstehen und diese miteinander vergleichen können.

- *Brüche mit einem Nenner von 10 und 100 im gleichen Zusammenhang verstehen, addieren und subtrahieren. Z.B $3/10 = 30/100$ oder $3/10 + 4/100 = 34/100$.*
- *Auch hier sollen die Brüche mit Hilfe der Zeichen $<$, $>$ und $=$ verglichen werden. Z.B $3/10 > 34/100$.*

Quelle dieser Angaben: (Massachusetts Curriculum Framework for Mathematics, 2011)

16.2.5 Messungen von Daten

In der Lage sein Problemstellungen im Bereich der Messung von Daten zu beantworten. Dazu gehören Umrechnungen und verschiedene Darstellungen der Messdaten.

- *Die bekannten Messgrößen wie km , m , cm ; t , kg , g ; lb , oz ; l , dl , ml ; hr , min , sec ; müssen vom Schüler beherrscht werden. Darüberhinaus muss der Schüler in der Lage sein Messungen in der jeweils nächstkleineren oder nächstgrösseren Einheit anzugeben. Zum Beispiel muss der Schüler wissen das 1 Liter aus 10 dl besteht und*

dazu Fragestellungen beantworten können. Z.B wie viele 1 dl Gläser können mit einem Liter gefüllt werden.

- *Mit Hilfe der vier Basisoperationen muss der Schüler in der Lage sein Textaufgaben betreffend Messgrößen wie Z.B Distanzen, Zeitintervalle, Volumen oder Geld zu lösen. Auch Textaufgaben bei denen Einheiten in kleinere Einheiten umgerechnet werden, müssen vom Schüler beherrscht werden.*
- *In Problemstellungen mit Rechtecken soll der Schüler in der Lage sein die fehlenden Werte zu berechnen. Z.B Gegeben ist ein Rechteck mit einer Fläche von 24m^2 und einer Breite von 8m^2 . Der Schüler muss die Länge des Rechtecks mit Hilfe der angegebenen Messdaten berechnen.*

Daten darstellen und interpretieren können.

- *Der Schüler muss die Daten in einem Diagramm darstellen können. Er muss befähigt sein Diagramme zu lesen und interpretieren.*

Quelle dieser Angaben: (Massachusetts Curriculum Framework for Mathematics, 2011)

16.2.6 Geometrie

In der Lage sein Linien und Winkel zeichnen und identifizieren zu können. Zudem müssen Formen anhand ihrer Eigenschaften, Linien und Winkel zugeordnet werden können.

- *Der Schüler weiß was Winkel sind und wie sich diese mit Hilfe eines Werkzeuges ablesen lassen. 90° Winkel beispielsweise muss er sofort ohne die Benützung eines Messgerätes erkennen.*
- *Ein Schüler muss wissen das ein Kreis 360° und ein Halbkreis 180° hat.*
- *Der Schüler muss in der Lage sein die Winkel gegebenenfalls zu addieren oder zu subtrahieren.*

Quelle dieser Angaben: (Massachusetts Curriculum Framework for Mathematics, 2011)

16.3 Anforderungen des Massachusetts Curriculum Framework for Mathematics in Bezug of OLPC

Untenstehende Tabelle zeigt verschiedene Lernziele. Zu jedem Lernziel wird angegeben ob es bereits eine bestehende Aktivität gibt welche das Lernziel deckt. Die Informationen stammen dabei von (Wiki Sugarlabs, 2014).

16.3.1 Zahlenverständnis und Umgang mit Operationen

Tabelle 4 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Zahlenverständnis und Basisoperationen. Quelle: (Wiki Sugarlabs, 2014)

Beschreibung	Wer arbeitet daran?	Welche Aktivität deckt das Bedürfniss?
Zahlenverständnis bis 100'000 fördern. Das Verständnis wird durch lesen, darstellen, schreiben und interpretieren der Zahlen gefördert. Die Zahlen werden miteinander verglichen und der Größe nach einsortiert.	Adam Schreiber	Deka, Abacus
Große Zahlen bis zu 100'000 darstellen und miteinander vergleichen. Dabei werden verschiedene Notationen wie z.B $780 = 7 * 100 + 2 * 20 + 18 + 22$.	UTOS-XO, Adam Schreiber	Deka
Demonstrieren das einzelne Teile Teil eines ganzen sind. Der Schüler soll verstehen, wie viel $\frac{3}{5}$ eines Kuchens sind.	Niemand	TurtleArt
Auswählen und gebrauchen von Modellen um Brüche zu verstehen. Z.B Brüche einordnen ($\frac{1}{2}$, $\frac{1}{3}$, $\frac{1}{4}$) und gleichbedeutende Brüche wie Z.B $\frac{2}{4}$ und $\frac{1}{2}$ erkennen.	Niemand	Abacus (Caacupé abacus) Worksheet, TurtleArt
Identifizieren und generieren von Kommazahlen und Brüchen. Z.B $\frac{8}{9} = 0.89$	Niemand	TurtleArt
Verstehen von Kommazahlen durch schreiben, lesen und gebrauchen von Hundertsteln zwischen 0 und 1.	Adam Schreiber	Deci, Turtle Art
Zahlen Klassen zuordnen. Z.B soll eine Zahl der Klasse	Doug Glenn	Keine

gerade, ungerade, Quadratzahl oder Produkt einer Multiplikation zugeordnet werden.		existierende Aktivität
Gebrauch und Erklärung verschiedener Multiplikationen und Divisionen von Ganzzahlen. Der Zusammenhang der Umkehrung dieser beider Operationen soll verstanden werden.	Steve Thomas	Etoys
Gebrauch vom Kommutativgesetz und Assoziativgesetz in realen Problemstellungen. Z.B. $37 * 46 = 46 * 37$, $(5*7)*2 = 5*(7*2)$	Niemand	Keine existierende Aktivität
Gebrauch der vier Grundoperationen Addition, Subtraktion, Multiplikation und Division in Problemstellungen mit Geldbezug.	Niemand	Keine existierende Aktivität
Multiplikations- sowie Divisionskenntnisse im Bereich $12*12$. Zusammenhänge zwischen $3*5$, $30 * 5$ und $300 * 5$ erkennen.	Keine Angaben	Keine Angaben
Addieren und subtrahieren von Zahlen im 5 stelligen Bereich und Multiplikation von dreistelligen mit zweistelligen Zahlen.	Niemand	Keine existierende Aktivität
Division von dreistelligen Ganzzahlen mit einstelligem Divisor.	Niemand	Keine existierende Aktivität
Runden von Zahlen bis zu 100'000. Die Zahlen sollen dabei auf die nächste 10er, 100er, 1'000er oder 10'000er Größe gerundet werden.	User:ossfm	Hop-A-Round
Einsetzen von verschiedenen Strategien, wie Z.B. Runden oder Gruppierungen um Resultate von bis zu dreistelligen Zahlen zu berechnen.	Niemand	Keine existierende Aktivität
Gebrauch von konkreten Modellen um Additionen und Subtraktionen auf Brüchen durchzuführen.	Niemand	Abacus (Caacupé abacus), Worksheet

16.3.2 Muster/Modelle, Relationen und Algebra

Tabelle 5 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Muster / Modelle, Relationen und Algebra. Quelle: (Wiki Sugarlabs, 2014)

Beschreibung	Wer arbeitet daran?	Welche Aktivität deckt das Bedürfniss?
Erstellen, erklären und erweitern von Mustern wie Z.B 8,80,800,8000....	Adam Schreiber, User:Dfu	Deka
Gebrauch von Symbolen und Zahlen zur Darstellung von Unbekannten Z.B x. in mathematischen Problemstellungen mit <,> und =.	Niemand	Keine existierende Aktivität
Lösungen in einfachen Gleichungen mit unbekannten ermitteln. Z.B $568-x=45$.	User:Gdk, User:Rholden	Dungeons of mongo
Gebrauch von Bildern, Modellen, Tabellen, Grafiken, Diagrammen, Wörtern, und mathematischen Notationen um mathematische Probleme darzustellen.	Niemand	Socialcalc Aktivität
Problemstellungen mit mengen- und Preisbezug. Z.B eine Zitrone kostet 0,80, wie viel Kosten sechs Zitronen? Ein Inch repräsentiert fünf Meilen daher repräsentieren zehn Meilen.	Niemand	Google Maps
Ermitteln wie eine Veränderung in einer Variablen eine andere beeinflusst. Z.B mit Hilfe von Input- Output Tabellen.	User:Rholden	None

16.3.3 Geometrie

Tabelle 6 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Muster / Geometrie. Quelle: (Wiki Sugarlabs, 2014)

Beschreibung	Wer arbeitet daran?	Welche Aktivität deckt das Bedürfniss?
Vergleichen und analysieren von verschiedenen Eigenschaften von Formen. Z.B Anzahl Seiten, Diagonalen, Symmetrie etc...	Niemand	Keine existierende Aktivität
Beschreiben, modellieren, zeichnen und vergleichen von zwei- und dreidimensionalen Formen.	Niemand	TurtleArt
Ähnliche Figuren erkennen	Niemand	Keine existierende Aktivität
Winkel als spitzen, stumpfen oder rechten Winkel einstufen können.	Christian Cook	Keine existierende Aktivität
Zeichnen und beschreiben von überschneidenden, parallelen und senkrechten Linien.	Niemand	Keine existierende Aktivität
Gebrauch von geordneten Zahlenpaaren, Graphen zur Beschreibung von Pfaden.	Matthew Daniels, Rob Seay	Nevhma (in Bearbeitung) TurtleArt
Beschreiben von Techniken wie Reflektion, Rotation um zu zeigen, dass zwei Applikationen identisch sind.	Niemand	TurtleArt
Identifizieren und beschreiben von symmetrischen Linien in zweidimensionalen Formen.	Niemand	Keine existierende Aktivität
Vorhersagen und validieren der Resultate für Partitionierung, Falten und Kombination von zwei und drei Dimensionalen Formen.	Niemand	Keine existierende Aktivität

16.3.4 Messungen

Tabelle 7 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Messungen. Quelle: (Wiki Sugarlabs, 2014)

Beschreibung	Wer arbeitet daran?	Welche Aktivität deckt das Bedürfniss?
Demonstrieren und Verstehen von Attributen wie Länge, Gewicht, Fläche etc...	Niemand	Keine existierende Aktivität
Einfache Umrechnungsfunktionen mit Messgrößen. Z.B Liter in Deziliter, Stunden in Minuten etc...	Niemand	Keine existierende Aktivität
Zeit anhand von analogen und digitalen Uhren ablesen. Fähigkeit vermitteln einen Kalender zu lesen.	Niemand	Clock Aktivität, GCompris, Clockgame, Stopwatch Aktivität, Turtle clock.
Fläche und Umfang von Vierecken, Dreiecken etc... mit Hilfe von Modellen berechnen können.	Niemand	Keine existierende Aktivität
Lernen mit dem Lineal, Geodreieck, Messzylinder oder Thermometer Messungen durchzuführen.	Niemand	Keine existierende Aktivität

16.3.5 Datenanalyse, Statistik und Wahrscheinlichkeit

Tabelle 8 : (Eigene Darstellung) Sugar Labs Math 4 Team TODO Datenanalyse, Statistik und Wahrscheinlichkeit.
Quelle: (Wiki Sugarlabs, 2014)

Beschreibung	Wer arbeitet daran?	Welche Aktivität deckt das Bedürfniss?
Sammeln, darstellen und organisieren von Daten durch Beobachten, Messen und experimentieren.	Niemand	Socialcalc Aktivität, TurtleArt sensor blocks
Aus gegebenen Daten eine Tabelle, Grafik oder ein Diagramm erstellen.	Niemand	Keine existierende Aktivität
Aus verschiedenen Daten sollen Rückschlüsse und Vorhersagen getroffen werden können. Mit Hilfe von Balkendiagrammen, Liniendiagrammen etc...	Niemand	Keine existierende Aktivität
Wahrscheinlichkeiten berechnen. Z.B Wie groß ist die Chance dass ich aus einem Topf mit 4 blauen und 3 roten Murmeln eine rote Murmel ziehe?	Niemand	Keine existierende Aktivität
Anzahl Möglichkeiten auflisten. Z.B wie viele verschiedene Outfits lassen sich mit zwei unterschiedlichen Hosen und drei unterschiedlichen T-Shirts erstellen.	Niemand	Keine existierende Aktivität
Ein Ereignis als wahrscheinlich, unwahrscheinlich oder unmöglich einstufen können.	Niemand	TurtleArt

16.4 Konkrete Softwareideen

Mit Hilfe obengenannter Ausführungen soll eine neue Applikation konzipiert werden. Mit Hilfe eines Brainstormings sollen neue Ideen gefunden werden. Untenstehendes Mindmap zeigt die Resultate des Brainstormings.

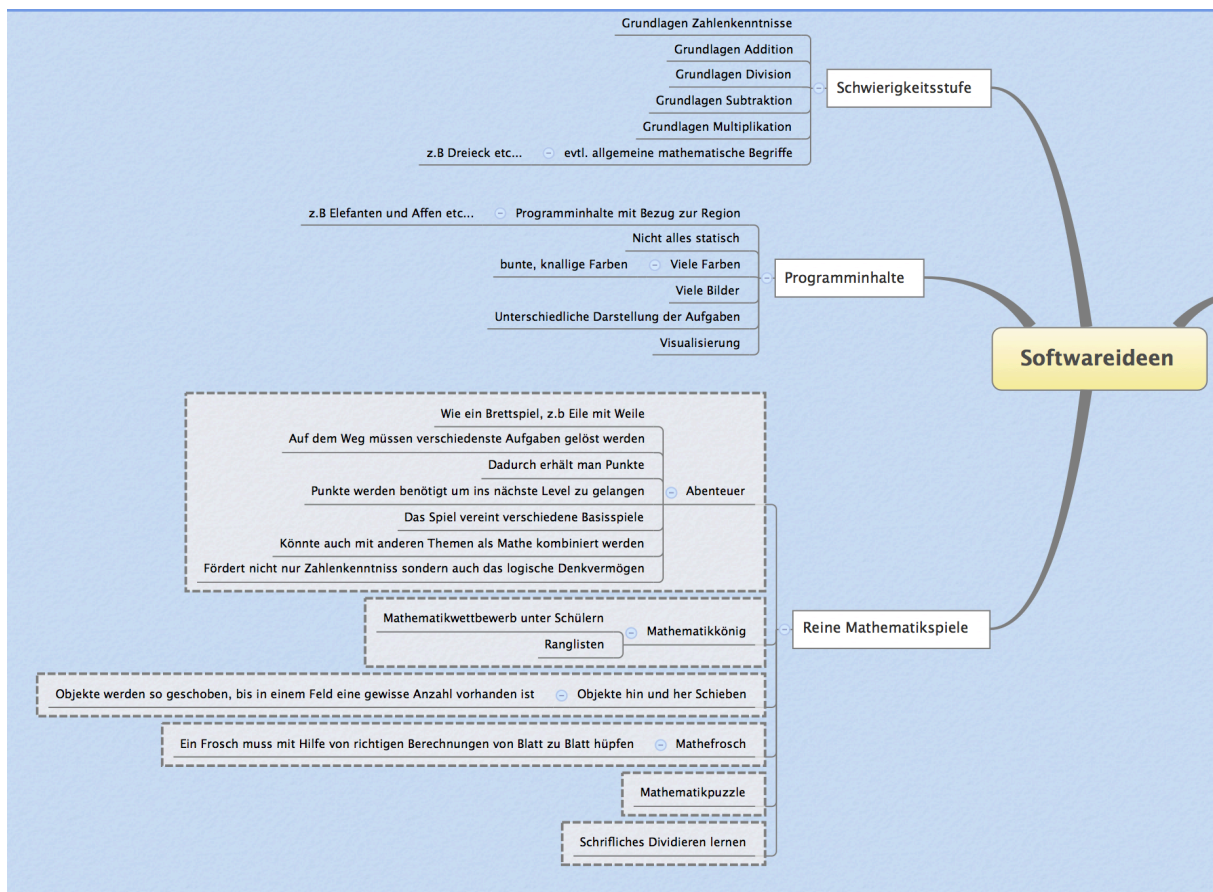


Abbildung 13 : (Eigene Darstellung) Brainstorming Teil 1

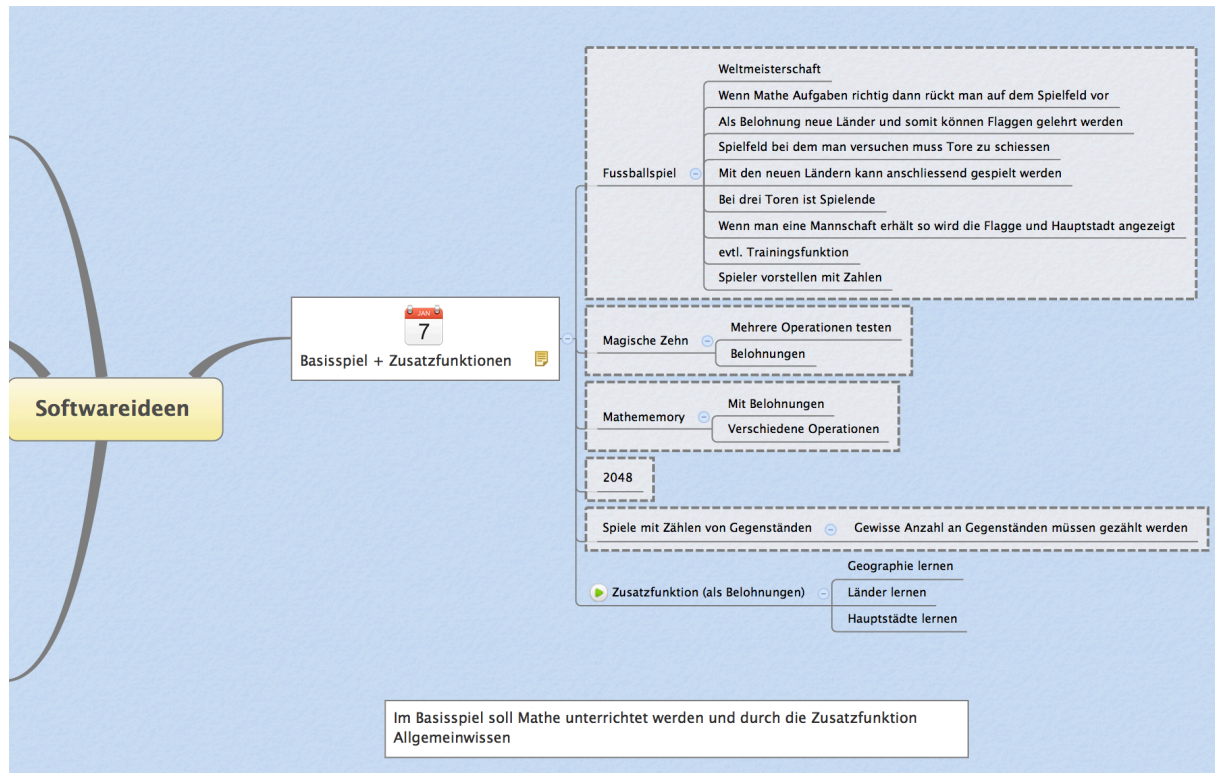


Abbildung 14 : (Eigene Darstellung) Brainstorming Teil 2

17 Sechster Entscheid - Bestimmen der Grundidee

Zu Beginn gefiel mir besonders die oben aufgelistete Idee mit dem Fußballspiel. Ziel dieses Spiels ist es ein Fußballspiel mit Einbezug mathematischer Aufgaben zu simulieren. Durch das Lösen von mathematischen Aufgaben könnte die Mannschaft den Ball nach vorne treiben und so anschließend Tore erzielen. Durch Siege werden dann neue Mannschaften freigeschaltet. Wird eine neue Mannschaft freigeschaltet, so wird die Flagge, der Name des Landes und die Landeshauptstadt angezeigt. Dies hätte den Vorteil, dass nebst Mathematik noch zusätzlich Geografie gelernt wird. Zudem könnte das Interesse der Kinder an einer solchen Applikation durch die Fussball Weltmeisterschaft bestärkt werden. Diese Idee habe ich auch mit meinen Experten Fabian Cretton und Laurent Sciboz ausführlich besprochen. Bei meinen Überlegungen habe ich nicht bedacht, dass diese Art von Spiel für Knaben sehr interessant, für Mädchen allerdings nicht wirklich spannend ist. Da besonders Mädchen über eine schlechte Grundausbildung verfügen gilt Ihnen ein besonderes Augenmerk. Aus diesem Grund suchte ich weiter nach einer geeigneten Spielidee.

Zu Beginn interessierten mich mehrere Spielideen. Allerdings waren die Spiele für den Rahmen meiner Bachelorarbeit zu klein. Hinzu kam, dass die Spielideen jeweils nur ein

kleines Bedürfnis decken. Zudem hatte ich meine Bedenken betreffend der Unterhaltsamkeit. Ich bin der Ansicht, dass einige der Spiele auf Dauer langweilig erscheinen können. Ich dachte mir also wie kann ich ein Spiel für Mädchen und Jungen abwechslungsreich und interessant gestalten, den Spielspass garantieren und einen möglichst hohen Lerneffekt, also viele Bedürfnisse abdecken?

Daraufhin betrachtete ich die auf dem Mindmap aufgelistete Idee des Abenteuerspiels. Dieses Basisspiel besteht aus einem Hauptspiel und verschiedenen kleinen Mathematik Minispiele. Die Oberfläche des Basisspiels gleicht dabei derjenigen eines Brettspiels. Durch Würfeln bewegt sich der Spieler auf dem Spielbrett vorwärts. Abhängig von der Position erscheint nach jedem Zug des Spielers ein kleines Minispiel mit Mathematikbezug. Durch die Vielfalt der Minispiele wird das Spiel abwechslungsreich und auch auf lange Dauer interessant. Weiter bieten die verschiedenen Minispiele dem Kind Möglichkeit mehrere Thematiken oder eine Thematik in verschiedenen Spielen zu erlernen. Diese Art des Spiels bietet mehr Abwechslung und einen höheren Lerneffekt. Wird ein Minispiel richtig beantwortet so erhält das Kind eine Belohnung. Mit Hilfe solcher Belohnungen können neue Levels freigeschaltet werden. Auch diese Idee wurde gemeinsam mit meinem Betreuer Laurent Sciboz besprochen. Herr Sciboz war ebenfalls von dieser Idee überzeugt. So beschlossen wir gemeinsam dieses Spiel im Rahmen meiner Bachelorarbeit zu realisieren.

18 Zusammenfassung der Entscheidungen

In meiner Analysephase wurden basierend auf obigen Informationen und Daten sechs Entscheide getroffen welche den Rahmen meiner Bachelorarbeit festlegen. Die Grundentscheide lauten wie folgt:

1. Die Applikation wird für grundsätzlich für einen OLPC Laptop implementiert. Durch die gewählte Web Technologie kann sie allerdings auch auf anderen Geräten laufen.
2. Die Applikation wird für spanisch sprechende Länder implementiert.
3. Der Fokus des Spiels liegt auf Mathematik. Es sollen besonders die Grundlagen der Mathematik trainiert werden. Auch andere Bedürfnisse können gedeckt werden.

4. Das Spiel wird als Sugar Web Aktivität umgesetzt. Das Spiel wird mit den Webtechnologien HTML, CSS und JavaScript umgesetzt. Selbstverständlich können verschiedene Bibliotheken wie JQuery oder zusätzliche Plugins benutzt werden.
5. Entwickelt wird auf einem Linux Fedora 19 „Schrödingers Cat“. Darauf wird mit osbuild eine volle Sugar Umgebung simuliert.
6. Das Spiel wird ein Abenteuerspiel. Der Spieler bewegt sich auf einem Brettspiel vorwärts und muss unterwegs verschiedene Mathematikaufgaben lösen. Durch das Lösen dieser Aufgaben kann der Spieler Bananen sammeln und so anschließend neue Levels freischalten.

Das Spiel

In diesem Teil der Arbeit wird das Spiel vorgestellt.

Er dient sogleich als Spielanleitung.

19 Aventura Matemática

Wie in der Zusammenfassung der Entscheidungen zu sehen ist, wird das Spiel eine Art „Abenteuerspiel“. Dieses Kapitel befasst sich ausschließlich mit dem Beschrieb des neuen Spiels.

19.1 Anforderungen

Die Anforderungen welche an eine solche Aktivität gestellt werden, können in Nichtfunktionale Musskriterien und Funktionale Muss- sowie Kann Kriterien unterschieden werden.

19.1.1 Nichtfunktionale Musskriterien:

- Zuverlässigkeit
- Benutzerfreundlich
- Look and Feel
- Codewartbarkeit - Die Aktivität sollte so geschrieben sein, dass sie gegebenenfalls von der Community erweitert werden kann.
- Effizienz
- Muss ein Lernbedürfnis decken
- Korrektheit
- Erkenntnistheoretischer Effekt – Es muss ein Lerneffekt erzielt werden
- Die Software muss abwechslungsreich und unterhaltsam sein
- Die Software muss qualitativ sein. Sie darf auf keinen Fall die Stabilität des Systems beeinträchtigen.
- „Sugarized“ – Die Aktivität soll in Bezug auf die Benutzerschnittstelle, das Journal, die Sicherheit und die Internationalisierung in Sugar integriert sein.
- Free Libre Open Source Software (FOSS) – Die Aktivität und alle Abhängigkeiten müssen frei und öffentlich zugänglich sein.
- Die Aktivität sollte in Sugar einzigartig sein.
- Die Aktivität soll den Erwartungen der Schüler und Lehrern entsprechen.
- Die Aktivität sollte einfach und schnell begreiflich sein.

Quelle: (Wikipedia.org)

19.1.2 Funktionale Musskriterien:

- Eine abwechslungsreiche Map (Level).
- Würfeln.
- Automatisches Vorwärtsbewegen des Spielers.
- Verschiedene Mathe Mini-Spiele müssen verfügbar sein. (Mindestens 3 verschiedene Spiele)
- Bei grünen Feldern +3 Bananen.
- Bei roten Feldern -3 Bananen.
- Bananen gegen Geld einlösen.
- Bei richtiger Beantwortung von Mathespielen müssen Bananen zum Bestand hinzugefügt werden.
- Bei falscher Beantwortung müssen Bananen abgezogen werden.
- Anzeige von aktuellem Feld + gewürfelte Zahl + Ergebnis.

19.1.3 Funktionale Kannkriterien:

- Evtl. mehrere Levels
- Noch mehrere Mini-Spiele
- Freischaltbare Elemente
- Einbauen von Musik
- Verschiedene Spielfiguren

19.2 Spielbeschreibung

Beim Spiel handelt es sich um ein Abenteuerspiel und ähnelt von der Grundidee her dem berühmten Spiel „Mario Party“. Die Oberfläche gleicht der eines Brettspiels. Durch einen simulierten Würfelwurf bewegt sich der Spieler auf dem Spielbrett vorwärts. In Abhängigkeit seiner Position erscheint anschließend ein Mini-Spiel mit Mathematikbezug. Jedes dieser Mini-Spiele deckt ein Bedürfnis und hilft dem Kind seine Mathematikgrundlagen zu verbessern. Löst das Kind die Aufgabe korrekt, so erhält es Bananen. Die Anzahl der erhaltenen Bananen sind abhängig vom jeweils gespielten Mini-Spiel. Ziel ist es möglichst viele Mini-Spiele richtig zu beantworten und so, so viele Bananen wie möglich zu sammeln. Fünfzig dieser Bananen können anschließend bei einem Affen gegen Gold eingetauscht

werden. Dies soll den Kindern das Prinzip des Tausches näher bringen. Schafft es ein Kind fünf Goldbarren zu sammeln, so wird das nächste Level freigeschaltet. Kern der Applikation ist die Hauptaktivität, das Spielfeld, welche durch verschiedenste mathematikbasierende Minispiele abgerundet wird. Untenstehendes Use Case verdeutlicht kurz die Interaktionen des Benutzers mit dem Spiel.

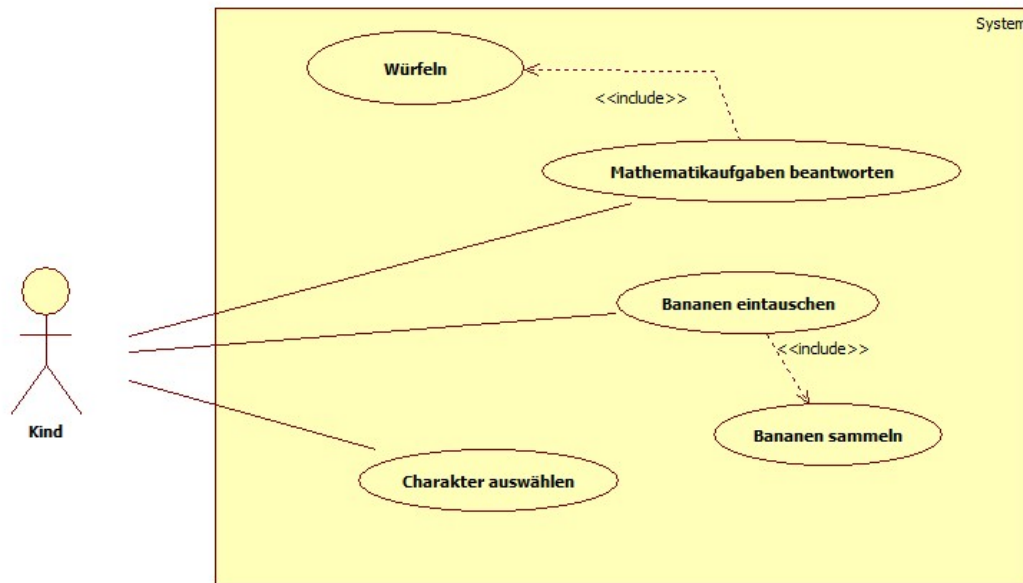


Abbildung 15 : (Eigene Darstellung) Use Case Adventura Mathematica

19.3 Ablauf einer Spielrunde

Wie bei einem Brettspiel hat auch dieses Spiel einen geregelten Ablauf einer Runde. Untenstehendes Sequenzdiagramm stellt die Interaktion und die verschiedenen Aufrufe dar.

1. Zuerst Würfelt das Kind durch einen Klick auf den Würfel
2. Die Spielfigur der Hauptaktivität bewegt sich anschließend vorwärts.
3. Nach der Bewegungsphase wird ein Mini-Spiel entsprechend der Farbe des Feldes der Spielfigur aufgerufen.
4. Das Kind kann nun das Mini-Spiel spielen.
5. Nach Beendigung des Mini-Spiels wird das Resultat übermittelt.
6. Nach der Übermittlung des Resultats werden die gewonnen oder verlorenen Bananen angezeigt.

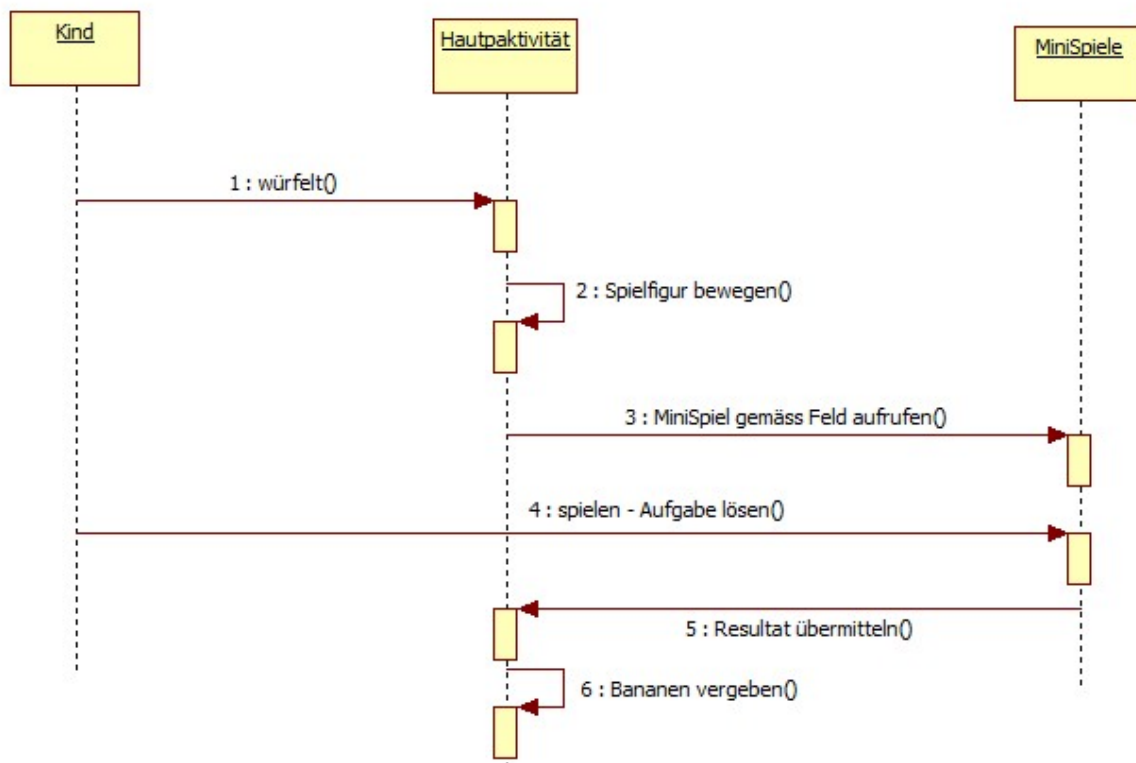


Abbildung 16 : (Eigene Darstellung) Sequenzdiagramm Ablauf Runde

19.4 Das Spiel beginnt!

Zu Beginn des Spiels erscheint dem Spieler folgender Auswahlbildschirm.



Abbildung 17 : (Eigene Darstellung) Auswahlbildschirm / Spielbeginn

Hier hat der Spieler die Möglichkeit die Anzahl der zu spielenden Runden zu wählen. Danach kann noch die Spielfigur und das Level ausgewählt werden. Auf obestehender Grafik ist zu

erkennen, dass sowohl das zweite als auch das dritte Level noch gesperrt sind. Diese können aber durch das Sammeln von jeweils drei Goldbarren im vorderen Level freigeschaltet werden. Auch der vierte Charakter ist gesperrt und muss durch das Sammeln von Bananen freigeschaltet werden. Auf dem Auswahlbildschirm befinden sich zudem noch drei Links. Der „Reglas del juego“ Link bietet eine Kurzübersicht über die Spielregeln. Der „Fuente“ Link enthält Backlinks zu einigen Bildquellen und mit Hilfe des „Restablecer puntuación“ können die gespeicherten Spielstände zurückgesetzt werden.

19.5 Hauptaktivität

19.5.1 Kurzbeschreibung und Übersicht

Bei der Hauptaktivität handelt es sich um das Spielbrett und den Kern des Spiels. Hier wird die Spielfigur vorwärts bewegt. Hier werden auch die Anzahl gesammelter Bananen und Goldbarren angezeigt.

19.5.2 Levels

In diesem Abschnitt werden die verschiedenen Levels kurz präsentiert. Die verschiedenen Levels wurden von mir selbst mit Hilfe von GIMP entworfen und erstellt. Die Bilder stammen dabei vorwiegend von openclipart.org. Um rechtlichen Problemen aus dem Weg zu gehen wurden alle verwendeten Bilder mit Hilfe von search.creativecommons.org gesucht. Bei jedem Level befindet sich der Würfel in der oberen linken Ecke. Daneben werden Ereignisse wie z.B die Vergabe von 10 Bananen angezeigt. Auch der Würfelwurf wird hier angezeigt. Die untere rechte Ecke zeigt eine Kurzübersicht über die bisher gesammelten Bananen und Goldbarren.

19.5.2.1 Level 1 – Isla Matemática

Untenstehende Abbildung zeigt ein Screenshot des ersten Levels der Applikation. Es handelt sich hierbei um einen Strand. Da es sich um das erste Level handelt werden hier Mini-Spiele mit Additions- und Subtraktionsaufgaben erscheinen.

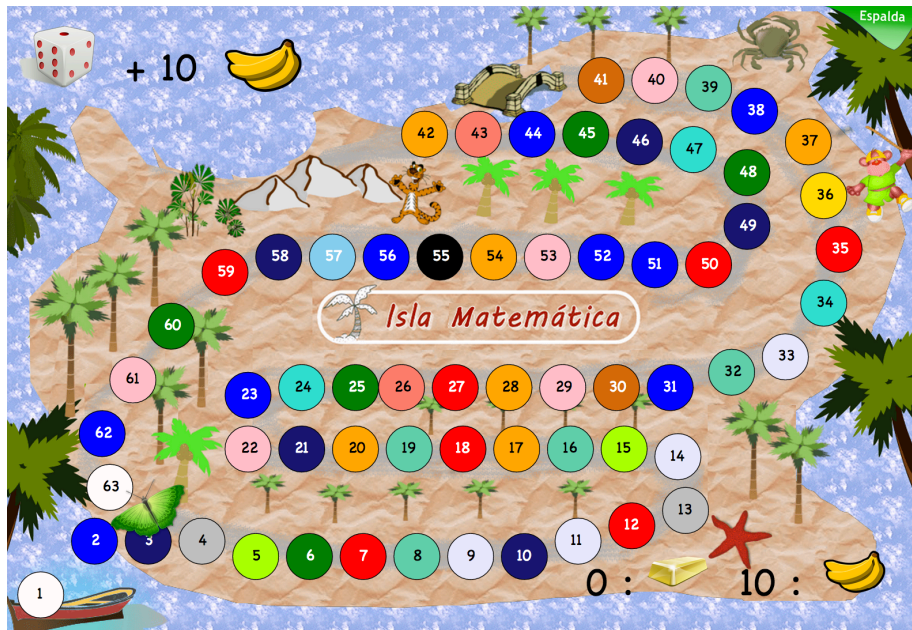


Abbildung 18 : (Eigene Darstellung) Screenshot Level 1

19.5.2.2 Level 2 – Universo Matemático

Die Abbildung illustriert das zweite Level. Zu Beginn des Spiels ist dieses Level noch gesperrt. Nachdem allerdings im ersten Level drei Goldbarren gesammelt wurden wird dieses zweite Level spielbar. In diesem Level werden im Gegensatz zum ersten Level wo noch Additions- und Subtraktionsaufgaben gestellt wurden, Multiplikations- und Divisionsaufgaben gestellt.

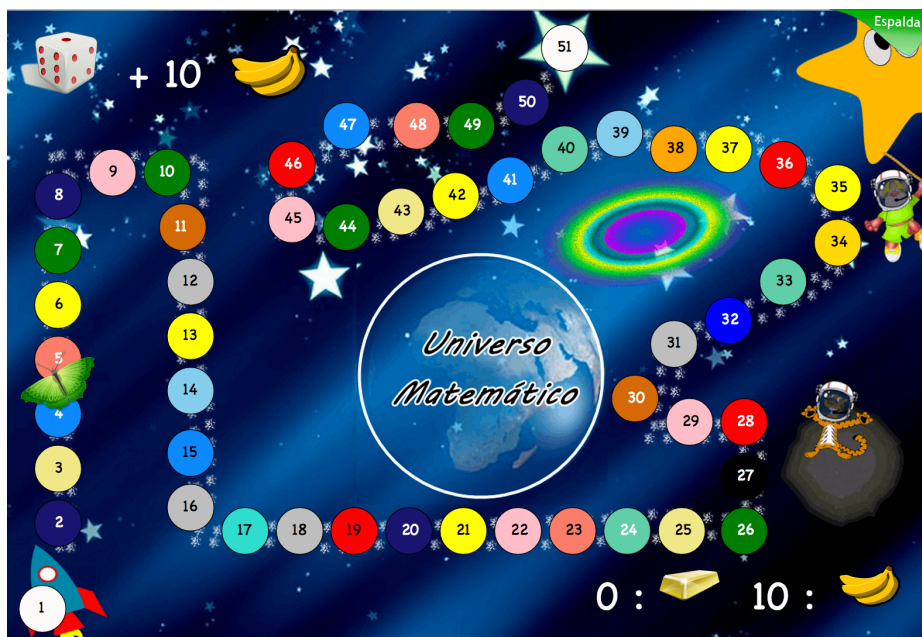


Abbildung 19 : (Eigene Darstellung) Screenshot Level 2

19.5.2.3 Level 3 – Viejo Oeste Matemático

Wie bereits das vorherige Level ist auch dieses Level bei Spielbeginn gesperrt. Das Level kann durch das Sammeln von drei Goldbarren im Level 2 freigespielt werden. Auch das dritte Level befasst sich hauptsächlich mit Multiplikation und Division. Es werden einzelne schwierigere Aufgaben gestellt als noch in den Leveln zuvor.

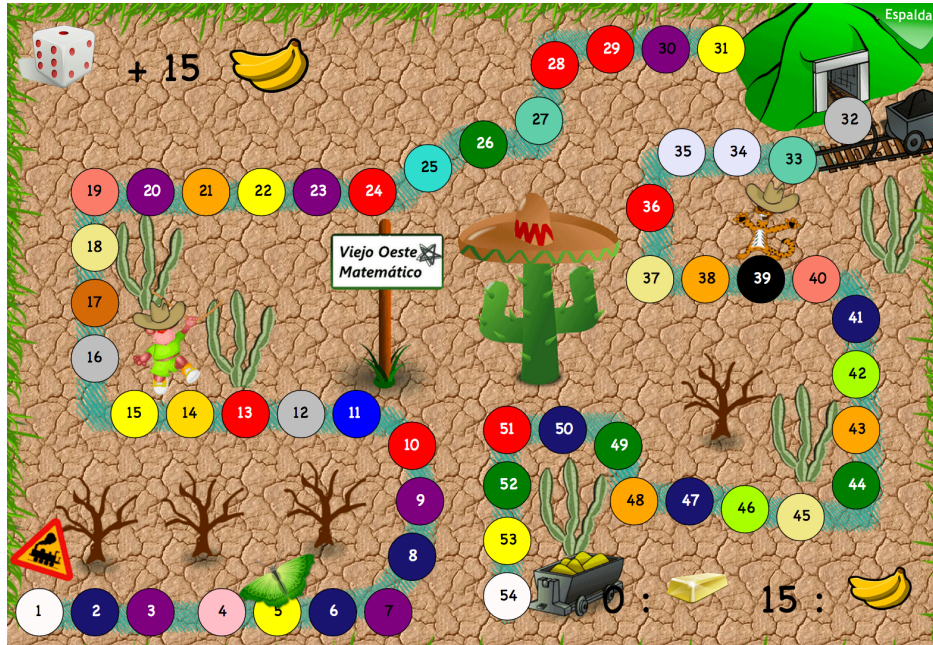


Abbildung 20 : (Eigene Darstellung) Screenshot Level 3

19.5.3 Use Case

Die Hauptaktivität verfügt nur über einen Use Case. Dieser Use Case ist für das Spiel aber von höchster Bedeutung.

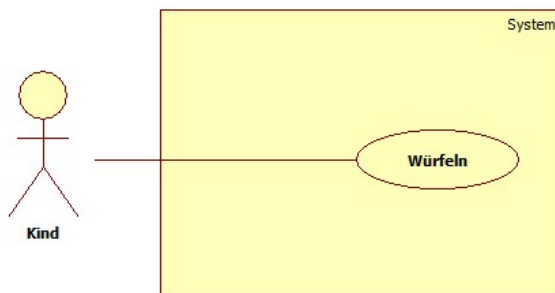


Abbildung 21 : (Eigene Darstellung) Use Case Hauptaktivität

19.5.3.1 Würfeln

Dieser Use Case ist besonders wichtig. Denn durch das Würfeln wird ein ganzer Fluss von Aktionen eingeleitet. Der ganze Ablauf einer Runde basiert auf dem Auslöser des Würfels.

Damit sich der Spieler auf dem Spielbrett fortbewegen kann muss er zuerst eine Zahl würfeln. Dazu stellt die Anwendung ein Würfelsymbol zur Verfügung. Dieses Würfelsymbol befindet sich immer in der oberen linken Ecke. Durch das Klicken dieses Würfelsymbols wird eine Zufallszahl generiert. Die generierte Zufallszahl wird dem Benutzer anschließend angezeigt. Da es sich um Kinder handelt und das Ziel der Applikation die Verbesserung von mathematischen Kenntnissen ist, soll die gewürfelte Zahl wie folgt angezeigt werden.

$$5 + \text{Würfel} = 8$$


Abbildung 22 : (Eigene Darstellung) Anzeige gewürfelte Zahl

Obenstehende Abbildung zeigt zuerst eine Fünf. Die Fünf ist gleichbedeutend mit dem Feld auf dem sich der Spieler aktuell befindet. Der Würfel zeigt die gewürfelte Zahl an. Die Acht steht für das Feld auf das der Spieler läuft. Durch diese Darstellung wird dem Kind bereits vermittelt, dass fünf plus drei acht ergibt. Damit dieses Wissen noch besser eingepreßt wird, ist auch die Bewegung der Spielfigur darauf abgestimmt. Die Spielfigur fliegt nicht einfach vom Feld fünf zum Feld acht. Die Figur nimmt jedes Feld Schritt für Schritt. Dies erlaubt es dem Kind jeden Schritt mitzuzählen und so zu erkennen, dass drei Felder vorrückt, nämlich die Felder sechs, sieben und acht.

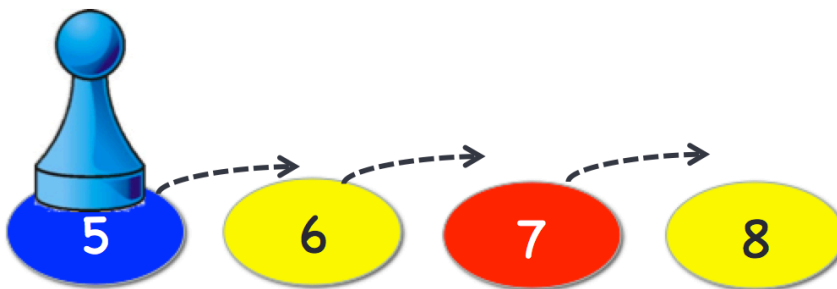


Abbildung 23 : (Eigene Darstellung) Vorwärtsbewegung des Spielers

Nachdem der Spieler seinen Zug gemacht hat, erscheint eine mathematische Aufgabe die das Kind anschließend zu beantworten versucht. Jede dieser mathematischen Aufgaben verfügt über ein neues Use Case Diagramm. Dieses Aufsplittern fördert die Übersicht und definiert für jedes einzelne Minispiel die Anforderungen.

19.6 Die Minispiele

19.6.1 Allgemeiner Überblick

Wie aus dem Sequenzdiagramm bereits zu erkennen ist, erscheinen die Mini-Spiele in Abhängigkeit mit der Farbe des Feldes auf dem die Spielfigur stehen bleibt. Folgende Tabelle gibt Auskunft darüber wann welches Mini-Spiel ausgeführt wird.

Tabelle 9 : (Eigene Darstellung) Mini-Spiele und die dazugehörige Farbe

Feldfarbe	Mini-Spiel
Blau	Shop Mini-Spiel
Gelb	Fischspiel
Orange	Formspiel
Lila	Endresultatspiel
Pink	Was kommt als nächstes?
Lavender	Käsespiel
Himmelblau	Zehnerspiel
Medium Aquamarine	Sortierspiel
Khaki	Slider-Spiel
Mitternachtsblau	Wandspiel
Türkis	Kreisspiel
Salmon	Mathematiktabelle
Grüngelb	Gitter Addition
Dodgerblau	Teilspiel
Chocolate	Memoryspiel
Silber	Schlossspiel

Einige der Mini-Spiele können in allen Levels auftauchen. Diejenigen Mini-Spiele welche in mehreren Levels erscheinen wurden so programmiert, dass sie sowohl Additionsaufgaben, Subtraktionsaufgaben als auch Multiplikationsaufgaben und Divisionsaufgaben beinhalten. Das Spiel bleibt dabei unabhängig von der Rechenart dasselbe. Die Rechnerart wird bei der Auswahl des Levels festgelegt. So kann auch innerhalb eines Levels die Rechenart ändern. Z.B wird im Level 1 vor allem Addition und Subtraktion trainiert. Dabei kann dasselbe Mini-

Spiel mehrmals erscheinen. Beim ersten Aufruf des Mini-Spiels wird dann beispielsweise eine Additionsaufgabe und beim nächsten Aufruf eine Subtraktionsaufgabe erscheinen. Einzelne Mini-Spiele können allerdings nur für eine der vier Grundrechenarten eingesetzt werden. Untenstehende Tabelle zeigt welche Mini-Spiele welche Bedürfnisse decken.

Tabelle 10 : (Eigene Darstellung) Nutzen-Wert der Mini-Spiele

Mini-Spiel	Nutzen-Wert
Shop-Spiel	Dieses Mini-Spiel fördert die Additionskenntnisse in Aufgaben mit Geldbezug.
Fischspiel	Dieses Mini-Spiel fördert die Multiplikationskenntnisse der Kinder. Weiter hilft es den Kindern zu erlernen die Zeichen $<$, $>$ und $=$ richtig einzusetzen.
Formen zählen	Dieses Mini-Spiel fördert die Kenntnisse von Formen. Kinder sollen damit lernen Formen des Alltags Begriffen wie Rechteck, Dreieck oder Kreis zuzuordnen. Weiter können die Kinder das Zählen trainieren.
Endresultatspiel	Dieses Mini-Spiel fördert die Addition und Subtraktionskenntnisse.
Was kommt als nächstes?	Bei diesem Mini-Spiel sollen logische Muster mit Hilfe von Formen erkannt werden. Muster wie z.B rot, blau, grün, rot, blau, grün etc...
Käsespiel	Bei diesem Spiel lernen die Kinder einfache Brüche zu addieren.
Zehnerspiel	Dieses Spiel soll den Kindern den Zusammenhang zwischen $4 * 10$, $4 * 100$, $4 * 1000$ etc.... verdeutlichen.
Sortierspiel	Bei diesem Spiel werden alle vier Grundrechenarten, also Addition, Subtraktion, Multiplikation und Division trainiert.
Slider-Spiel	Dieses Spiel befasst sich wiederum mit Brüchen. Ziel ist es das die Kinder einen abgebildeten Bruch auf einem Strahl als Dezimalzahl darstellen können.
Wandspiel	Dieses Spiel trainiert sowohl Addition, Subtraktion, Division als auch Multiplikation.
Kreissspiel	Mit Hilfe dieses Spiels sollen die Kinder lernen einen Kreisausschnitt in einem Kreis einzuzeichnen. Es soll vermittelt werden, dass ein Kreis z.B einen 360° und ein Halbkreis z.B einen 180° Winkel hat.
Mathematiktafel	Dieses Spiel fördert wiederum die vier Grundoperationen Addition,

	Subtraktion, Multiplikation und Division
Gitter Addition	Wie der Name schon sagt befasst sich dieses Mini-Spiel mit dem Training der Addition. Zudem sollen die Kinder erkennen wie viele Klötzchen eines Blockes eingefärbt sind.
Teilspiel	Dieses Spiel optimiert die Divisionskenntnisse.
Memoryspiel	Das Memoryspiel trainiert alle vier Grundrechenarten.
Schlossspiel	Auch das Schlossspiel trainiert alle vier Grundrechenarten.

Die Mini-Spiele erscheinen wie folgt in den verschiedenen Levels.

Tabelle 11 : (Eigene Darstellung) Vorkommnisse der Mini-Spiele in den Levels

Mini-Spiel	Level 1	Level 2	Level3
Shopspiel	✓		
Fischspiel		✓	✓
Formen zählen	✓		✓
Endresultatspiel			✓
Was kommt als nächstes?	✓	✓	✓
Käsespiel	✓		✓
Zehnerspiel		✓	
Sortierspiel	✓	✓	✓
Slider-Spiel		✓	✓
Wandspiel	✓	✓	✓
Kreisspiel	✓	✓	✓
Mathematiktafel	✓	✓	✓
Gitter Addition	✓		✓
Teilspiel		✓	
Memoryspiel	✓	✓	✓
Schlossspiel	✓	✓	✓

19.6.2 Der MiniShop

19.6.2.1 Spielbeschreibung und Übersicht

Aus der Analyse der Bedürfnisse, geht hervor, dass Aufgaben mit Geldbezug fehlen. Zurzeit arbeitet niemand an einer solchen Aktivität und es existiert derzeit auch keine für den OLPC Laptop. Aus diesem Grund wird sich dieses Mini-Spiel mit Geld befassen. Das Mini-Spiel spielt sich in einem kleinen Shop ab. Der Spieler erhält zu Beginn des Mini-Spiels einen zufälligen Betrag an Geld. Im Shop muss er anschließend die ausgestellten Sachen so einkaufen, dass sein Einkaufstotal genau mit der Menge des Geldes übereinstimmt.

Compra artículos haciendo clic sobre ellos. Intenta alcanzar el valor indicado.



Abbildung 24 : (Eigene Darstellung) Minispiel Shop

Das Regal auf der linken Seite zeigt fünf verschiedene Elemente. Einen Basketball, eine Seife, eine Gitarre, einen Ball und einen Teddybär. Neben dem Element ist der jeweilige Preis angegeben. Auf der rechten Seite ist ein Warenkorb mit den bereits angeklickten Elementen dargestellt. Alle Elemente die sich im Korb befinden werden darunter aufgelistet. Die Elemente müssen so in den Warenkorb gelegt werden, dass ihre Summe mit dem Total übereinstimmt. Für dieses Minispiel ergeben sich daher folgende Use Cases.

19.6.2.2 Use Cases

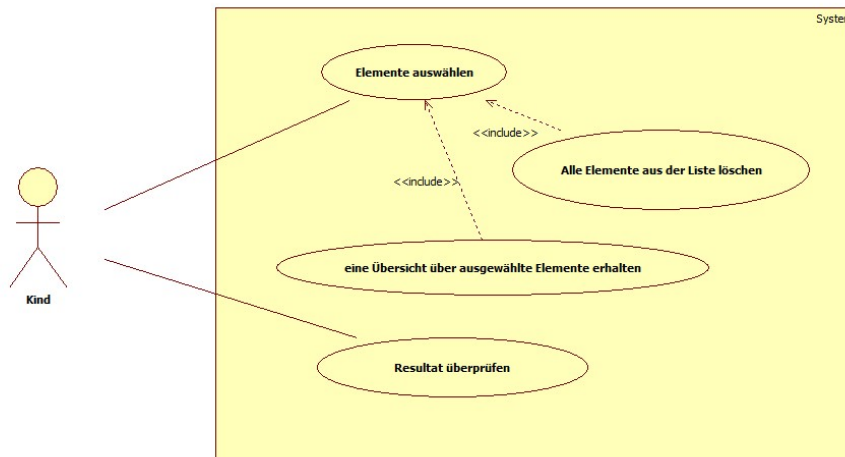


Abbildung 25 : (Eigene Darstellung) Use Case Minispiel Shop

Elemente auswählen

Auf der graphischen Benutzeroberfläche des Mini-Spiels „Shop“ findet der Nutzer auf der linken Seite ein Regal mit verschiedenen Elementen. Um eines dieser Elemente auszuwählen, genügt ein einfacher Klick auf eines dieser Elemente.

Übersicht über ausgewählte Elemente erhalten

Sobald ein Element angeklickt wird, erscheint dieses sowohl im Warenkorb als auch in der Einkaufsliste. Während im Warenkorb das Bild erscheint wird in der Einkaufsliste der Name und der Betrag angegeben. Untenstehende Abbildung zeigt sämtliche Auflistungen der selektierten Elemente.

Alle Elemente aus der Liste löschen

Mit Hilfe des Button „Vaciar“ können die Elemente aus der Liste gelöscht werden. „Vaciar“ löscht alle Elemente aus der Einkaufsliste und gleichzeitig auch alle Bilder aus dem Warenkorb.

Resultat überprüfen

Um das Resultat zu überprüfen dient der Button „Pagar“. Stimmt die Summe der selektierten Elemente mit dem Total überein so wird die Aufgabe als korrekt gewertet. Nach Beendigung dieser Überprüfung kehrt man zum Hauptspiel zurück. Bei der richtigen Beantwortung der

Fragestellung erhält das Kind 10 Bananen. Bei falscher Beantwortung wird der Bestand an Bananen um 5 verringert.

19.6.3 Fisch-Spiel

19.6.3.1 Spielbeschreibung und Übersicht

Mit Hilfe dieses Spiels sollen die Kinder in erster Linie lernen die Zeichen grösser als, kleiner als und gleich groß zu gebrauchen. Nebst dem Vergleich von Mengen wird das ein mal eins trainiert. Der Schwierigkeitsgrad beschränkt sich auf den Bereich von 12 x 12. Untenstehende Abbildung zeigt wie die graphische Schnittstelle aussieht.



Abbildung 26 : (Eigene Darstellung) Graphische Schnittstelle Fischspiel

In der Grafik sind zwei Rechtecke zu erkennen. In jedem dieser Teiche schwimmen eins bis zu 12 Fische. Neben jedem der Rechtecke ist eine Zahl angegeben (hier neun mal und sieben mal). Die Aufgabe des Kindes besteht nun darin, die Anzahl Fische des ersten Teiches mit der danebenstehenden Zahl zu multiplizieren und auszurechnen. Dieselbe Rechnung muss für den linken Teich durchgeführt werden. Danach muss das Kind mit Hilfe der obenstehenden drei Buttons entscheiden ob das Resultat der Multiplikation von Teich eins grösser, kleiner oder gleich groß ist wie dasjenige des Teichs zwei. Sobald der Fisch gezeichnet wurde, erscheinen die richtigen Resultate der Multiplikationen mit dem dazugehörigen Zeichen. Ein blaues „Smiley“ gibt Auskunft über die Richtigkeit der Antwort. Mit dem unteren Button kann man zur nächsten Frage übergehen.

19.6.3.2 Use Cases

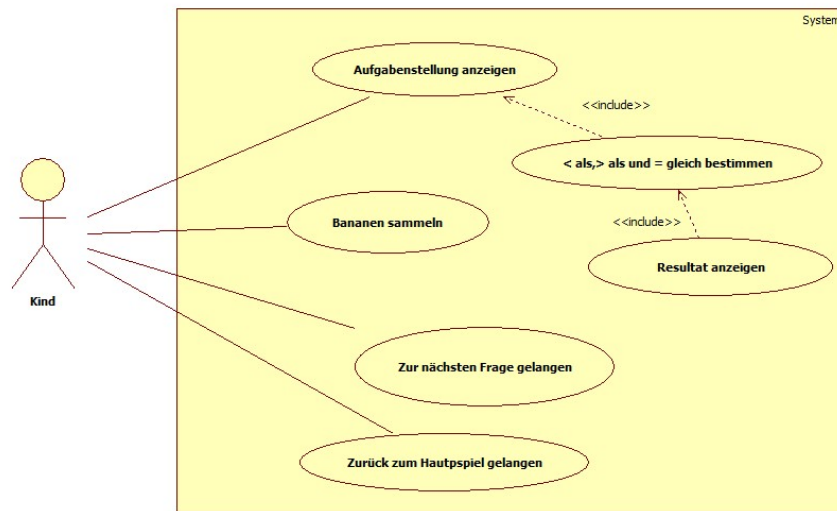


Abbildung 27 : (Eigene Darstellung) Use Case Fischspiel

Aufgabenstellung anzeigen

Das Kind erwartet, dass das System die Aufgabenstellung verständlich und einfach angezeigt wird. Die Aufgabenstellung wird direkt beim Start der Applikation dargestellt. Die Aufgabenstellung stellt sich aus zwei Multiplikatoren und zwei Teichen mit Fischen zusammen. Die Anzahl Fische im Teich müssen mit den Multiplikatoren multipliziert werden. Das Kind muss beide Resultate errechnen (z.B 8 mal 5 und 3 mal 10). Im nächsten Use Case entscheidet das Kind ob eines der beiden Resultate grösser, kleiner oder ob beide gleich groß sind.

„> als, < als oder gleich groß wie“ bestimmen

Nachdem das Kind die beiden Resultate der Multiplikationen errechnet hat, muss es bestimmen, welches dieser beiden grösser oder kleiner ist. Dies kann mit Hilfe der drei Buttons (>, = und <) bestimmt werden.

Nach dem Klick auf eines der drei Buttons wird zur Visualisierung ein Fisch gezeichnet. Dieser Fisch soll den Kindern helfen die Bedeutung der Symbole zu verinnerlichen. Das Maul des Fisches wird in Abhängigkeit zum gedrückten Button gezeichnet. Die Kinder sollen sich

merken, dass der Fisch in der Mitte immer in dem Rechteck frisst, in dem es mehrere Fische hat.

Resultat anzeigen

Nachdem einer der drei Buttons gedrückt wurde, also der Fisch gezeichnet wurde, wird das Resultat angezeigt. Es werden sowohl beide Resultate der Multiplikation sowie das zugehörige Zeichen angezeigt. Ein zusätzliches Smiley neben der Antwort zeigt an ob die Frage richtig oder falsch beantwortet wurde.

Zur nächsten Frage und zurück zum Hauptspiel

Mit Hilfe des Buttons am unteren Bildrand kann das Kind zur nächsten Frage wechseln. Nachdem drei Fragen gestellt wurden, wechselt der Button automatisch auf „Continuar“. Mit Hilfe dieses Buttons gelangt man zurück zum Hauptspiel.

Bananen sammeln

Wie in jedem Mini-Spiel können auch hier Bananen gesammelt werden. Pro Runde erscheinen jeweils drei Aufgabenstellungen. Am Ende des Spiels werden die Bananen gemäß richtiger Antworten verteilt. Untenstehende Tabelle zeigt die vergabe der Anzahl Bananen.

Tabelle 12 : (Eigene Darstellung) Fischspiel - Vergabe der Bananen

Anzahl richtiger Antworten	Anzahl Bananen
0	-10
1	5
2	10
3	15

19.6.4 Formen zählen

19.6.4.1 Spielbeschreibung und Übersicht

Ziel dieses Spiels ist es das die Kinder Begriffe wie Kreis, Rechteck oder Dreieck alltäglichen Gegenständen zuordnen können. So werden in diesem Spiel verschiedenste Objekte aus dem

Alltag dargestellt. Das Kind muss anschließend angeben wie viele Dreiecke, Vierecke und Kreise abgebildet sind.

Cuenta el número de veces que aparece cada forma





















				Rectanglos	<input type="text" value="5"/>
				Circulos	<input type="text" value="11"/>
				Trianglos	<input type="text" value="4"/>
					
				Vaciar	Verificar

Abbildung 28 : (Eigene Abbildung) Interface Spiel Formen zählen

19.6.4.2 Use Cases

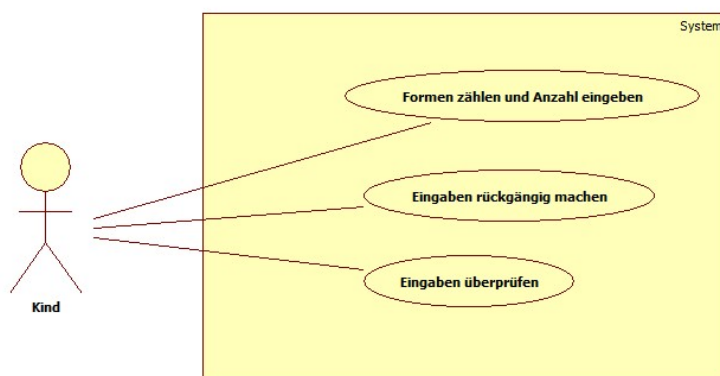


Abbildung 29 : (Eigene Abbildung) Use Case Spiel Formen zählen

Formen zählen und Anzahl eingeben

Wie im Spielbescrieb zu sehen ist werden verschiedenste Formen aus dem Alltag dargestellt. Das Kind hat zur Aufgabe diese zu zählen und anschließend die Anzahl der Dreiecks-, Rechtecks- und Kreisformen in die dafür vorgesehenen Felder einzutragen.

Eingaben rückgängig machen

Um die Eingaben rückgängig zu machen muss der Button „Vaciar“ geklickt werden. Dieser löscht alle Einträge aus den Eingabefeldern.

Eingaben überprüfen

Durch die Betätigung des „Verificar“ Buttons werden sämtliche Eingaben überprüft. Bei richtiger Beantwortung erhält das Kind 10 Bananen. Ansonsten verliert es 10 Bananen.

19.6.5 Plus und Minus einsetzen

19.6.5.1 Spielbeschreibung und Kurzübersicht

Bei diesem Spiel werden die Additions- und Subtraktionsfähigkeiten des Kindes trainiert. Sowohl Additionen und Subtraktionen aus denen eine positive Zahl resultiert als auch Additionen und Subtraktionen mit einem negativen Endergebnis werden trainiert. Dieses Mini-Spiel zählt zu den schwersten Mini-Spielen und erscheint deshalb erst in Level 3.

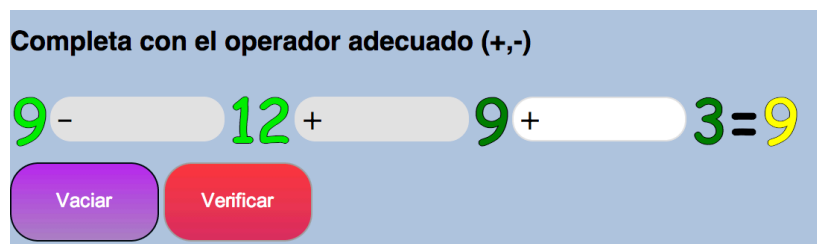


Abbildung 30 : (Eigene Darstellung) Grafische Oberfläche Endresultat Mini-Spiel

Das Kind muss die Zeichen + und – so einsetzen, dass die Rechnung stimmt.

19.6.5.2 Use Cases

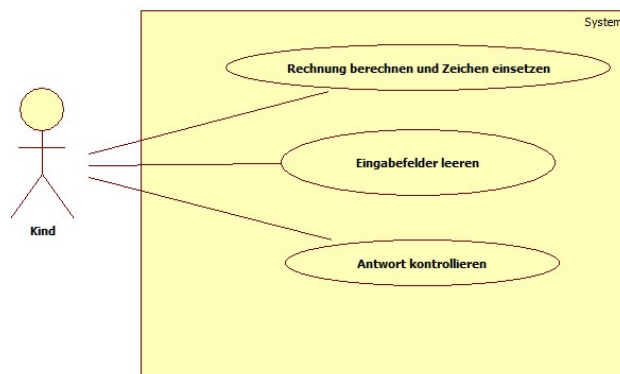


Abbildung 31 : (Eigene Darstellung) Use Case Endresultat Mini-Spiel

Rechnung berechnen und Zeichen einsetzen

Die Rechnung wird Anhand von zufälliger Auswahl der Zeichen und Zahlen konstruiert. Das Kind muss nun die Zeichen so einsetzen, dass die Rechnung schlussendlich stimmt. Die Zeichen plus und minus können in die jeweiligen Eingabefelder eingegeben werden.

Eingaben leeren

Um die Eingabefelder zu leeren muss der Button „Vaciar“ geklickt werden. Dieser löscht alle Einträge aus den Eingabefeldern.

Eingaben überprüfen

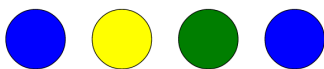
Durch die Betätigung des „Verificar“ Buttons werden sämtliche Eingaben überprüft. Stimmen alle Zeichen so erhält das Kind 20 Bananen. Die Bananenvergabe fällt aufgrund des Schwierigkeitsgrades so hoch aus. Ist die Aufgabe falsch, stimmt also bereits nur ein Zeichen nicht, so verliert das Kind 5 Bananen.

19.6.6 Muster mit Hilfe von Formen erkennen

19.6.6.1 Kurzbeschreibung und Übersicht

Ziel dieses Spiels ist es, dass das Kind lernt Muster zu erkennen. Muster in diesem Zusammenhang bedeutet, Abfolgen welche durch logisches Denken erkennbar sind. Die obere Zeile stellt das Muster und die untere die jeweiligen Elemente dar.

Cuenta el número de veces que aparece cada forma



Cuenta el número de veces que aparece cada forma



Abbildung 32 : (Eigene Darstellung) Grafische Schnittstelle „Was kommt als nächstes?“

19.6.6.2 Use Cases

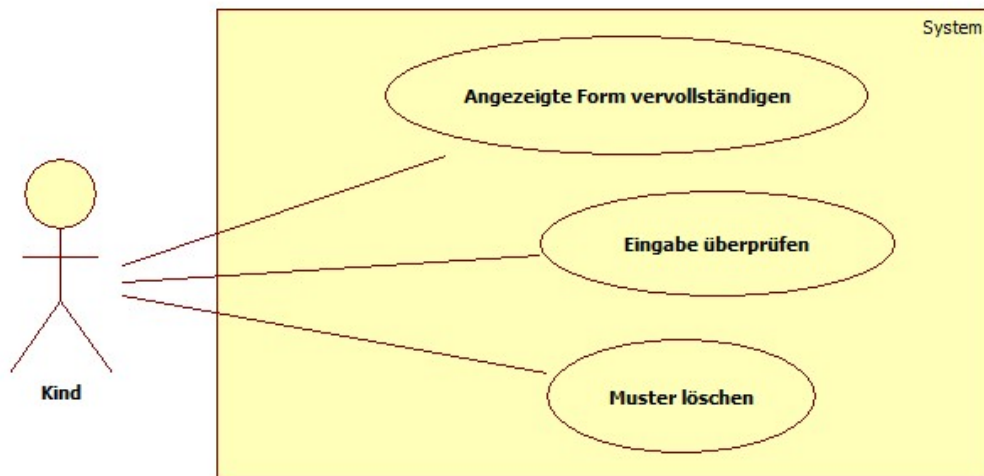


Abbildung 33 : (Eigene Darstellung) Use Case „Was kommt als nächstes?“

Angezeigte Form vervollständigen

Mit Hilfe der verfügbaren Elemente liegt es am Kind eine logische Struktur im angegebenen Muster zu entdecken und zu vervollständigen.

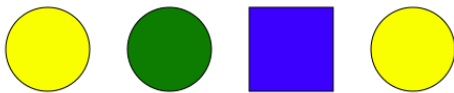


Abbildung 34 : (Eigene Darstellung) Beispiel eines Musters

Dieses Muster beispielsweise wird mit einem grünen Kreis, einem blauen Rechteck und wiederum einem gelben Kreis weitergeführt. Das Muster muss solange vervollständigt werden bis 10 Formen gezeichnet worden sind.

Muster löschen

Falls man bei der Eingabe des Musters einen Fehler macht hat man durch Betätigen des „Borrado“ Buttons die Möglichkeit die bisher gemachten Eingaben zu löschen und neu zu erfassen.

Eingabe überprüfen

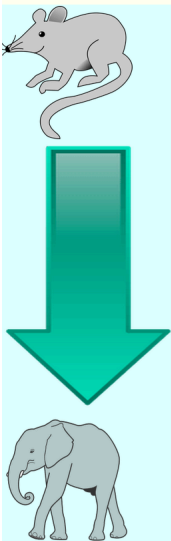
Nach Eingabe von 10 Formen wird das eingegebene Muster automatisch überprüft. Bei der Eingabe eines richtigen Musters erhält das Kind 10 Bananen plus. Ansonsten werden 10 Bananen vom bisherigen Bestand subtrahiert.

19.6.7 Sortiert Mini-Spiel

19.6.7.1 Kurzbeschreibung und Übersicht

Mit Hilfe dieses Spiels sollen die Kenntnisse in den Grundoperationen verbessert werden. Das Kind sortiert Rechnungen anhand ihres Resultates ein. Zuerst werden die Rechnungen mit dem kleinsten Resultat einsortiert. Der Algorithmus wurde dabei so geschrieben, dass nur eindeutige ganzzahlige Ergebnisse vorkommen.

Ordena los elementos comenzando por el más pequeño



$6 - 2$
$12 - 6$
$8 - 1$
$12 - 4$
$11 - 1$
$14 - 1$
$18 - 2$

Abbildung 35 : (Eigene Darstellung) Sortiert Mini-spiel Benutzeroberfläche

19.6.7.2 Use Cases

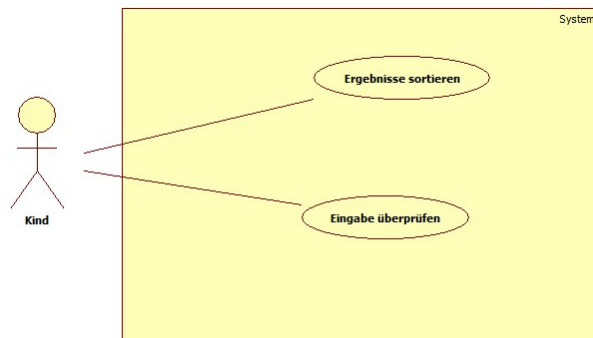


Abbildung 36 : (Eigene Darstellung) Use Case Sortiert Mini-Spiel

Ergebnisse sortieren

Das Kind kann die Ergebnisse einfach per Drag & Drop sortieren. Um beispielsweise das dritte Element der Liste ganz oben zu positionieren muss dieses mit gedrückter Maustaste über das erste Element gezogen werden. Beim loslassen der Maustaste sortieren sich die Elemente anschließend von selbst ein.

Eingabe überprüfen

Mit dem Button „Verificar“ können die Eingaben überprüft werden. Sind alle Elemente korrekt sortiert, so erhält das Kind 10 Bananen. Andernfalls verliert es 10 Bananen.

19.6.8 Zehnerspiel Mini-Spiel

Dieses Spiel soll dem Kind die Zusammenhänge zwischen beispielsweise $2 \cdot 10$, $2 \cdot 100$, $2 \cdot 1'000$ und $2 \cdot 10'000$ vermitteln. Das Kind soll verstehen dass für jede zusätzliche Null im Multiplikator eine zusätzliche Null zum Resultat hinzugefügt werden muss. Das Kind muss für jedes Ergebnis die Anzahl Nullen angeben.

Añade el número correcto de ceros a cada resultado.

$$\begin{array}{rcl}
 36 * 1 & = & 36 \text{ ? ? ? ? } \\
 36 * 10 & = & 36 \text{ 0 ? ? ? } \\
 36 * 100 & = & 36 \text{ 0 0 ? ? } \\
 36 * 1000 & = & 36 \text{ 0 0 0 ? } \\
 36 * 10000 & = & 36 \text{ 0 0 0 0 }
 \end{array}$$

Verificar

Abbildung 37 : (Eigene Darstellung) Zehnerspiel Benutzeroberfläche

19.6.8.1 Use Cases

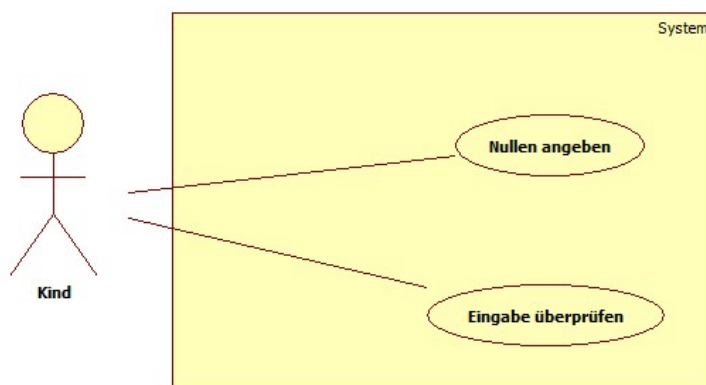


Abbildung 38 : (Eigene Darstellung) Use Case Zehnerspiel

Nullen angeben

Ziel des Spiels ist es, dass das Kind die Nullen am Ende jedes Resultates richtig angibt. Mit einem einfachen Klick kann ein Fragezeichen zu einer Null geändert werden.

Eingabe überprüfen

Mit dem „Verificar“ Button können die Eingaben überprüft werden. Bei einer korrekten Antwort erhält das Kind 10 Bananen. Bei einer falschen Antwort verliert es 10 Bananen.

19.6.9 Wand Mini-Spiel

19.6.9.1 Kurzbeschreibung und Übersicht

Bei diesem Spiel wird das Rechnen mit den Grundoperationen trainiert. Auf der graphischen Benutzeroberfläche ist eine Wand abgebildet welche dem Panda den Weg versperrt. Durch lösen der Rechnung kann die Wand nacheinander entfernt und so dem Panda der Weg freigemacht werden.

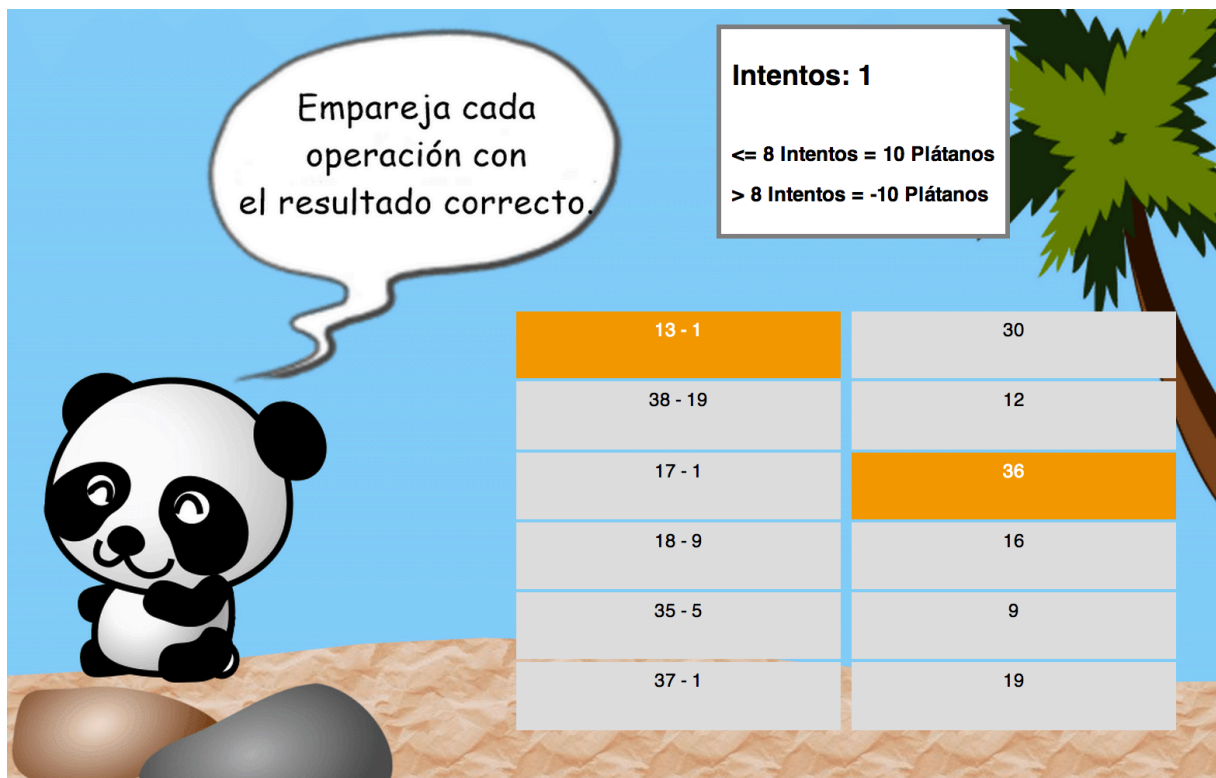


Abbildung 39 : (Eigene Darstellung) Graphische Benutzeroberfläche Wand Mini-Spiel

19.6.9.2 Use Cases

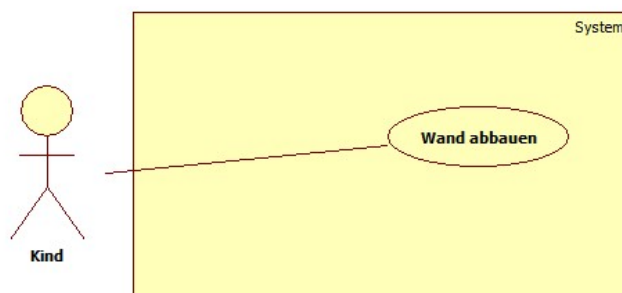


Abbildung 40 : (Eigene Darstellung) Use Case Diagramm Wand Mini-Spiel

Eingabe überprüfen

Die Benutzeroberfläche zeigt zwei Blöcke. Der Block links beinhaltet die Aufgabenstellung z.B. $16+44$. Der rechte Block beinhaltet alle Antworten. Ziel ist es nun die Aufgabe im linken Block mit der richtigen Lösung im rechten Block zu verbinden. Dazu muss die richtige Aufgabenstellung und die richtige Lösung angeklickt werden. Sobald die ausgewählte Aufgabenstellung und das ausgewählte Resultat übereinstimmen, so verschwinden diese Blockteile. Dadurch kann die Mauer schrittweise abgebaut werden. Ist die Mauer vollständig abgebaut, so kehrt man zum Hauptspiel zurück und erhält 10 Bananen plus.

19.6.10 Slider Mini-Spiel

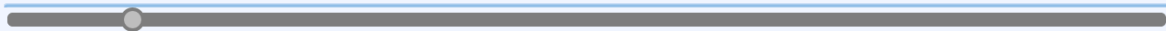
19.6.10.1 Kurzbeschreibung und Übersicht

Ziel dieses Spiels ist es den Umgang mit Brüchen zu trainieren. Besonders aber auch den Zusammenhang zwischen Brüchen und Ganzzahlen zu erkennen. Das $\frac{1}{2}$ beispielsweise 50 von 100 sind. Das Kind soll auf einem Slider der sich hin und her bewegen lässt einen Bruch darstellen. Das Minimum des Sliders beträgt dabei 0 und das Maximum 100.

Arrastra el punto para representar la fracción

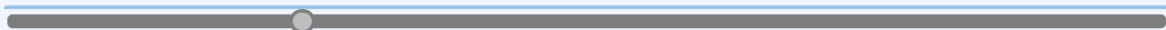
1/10

Valor : 10



1/4

Valor : 25



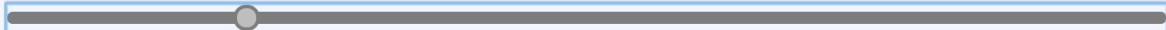
5/10

Valor : 50



1/5

Valor : 20



Verificar

Abbildung 41 : (Eigene Darstellung) Grafische Benutzeroberfläche Slider Mini-Spiel

19.6.10.2 Use Cases

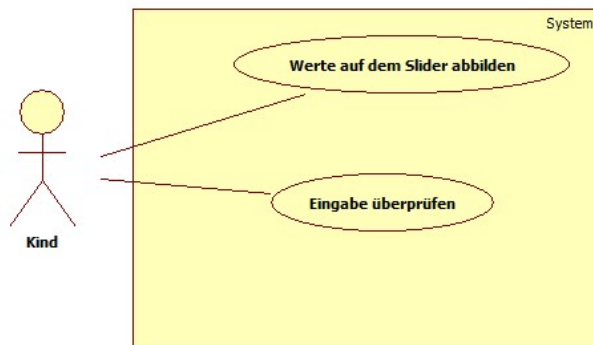


Abbildung 42 : (Eigene Darstellung) Use Case Slidergame

Werte auf dem Slider abbilden

Die Werte können durch einfaches verschieben des Reglers an jedem Sliders abgebildet werden. Die Anzeige „Valor“ direkt über dem Slider gibt Auskunft darüber wie weit der Regler auf einer Skala von 0 bis 100 gezogen wurde.

Eingaben überprüfen

Mit dem „Verificar“ Button können die Eingaben überprüft werden. Bei einer korrekten Antwort erhält das Kind 10 Bananen. Bei einer falschen Antwort verliert es 10 Bananen.

19.6.11 Vervollständige den Käse“ Mini-Spiel

19.6.11.1 Kurzbeschreibung und Übersicht

Bei diesem Spiel wird der Umgang mit Brüchen gefördert. Das Kind soll Stücke eines Käses so einsetzen, dass der Käse vervollständigt wird.



Abbildung 43 : (Eigene Darstellung) Käsespiel Benutzeroberfläche

19.6.11.2 Use Cases

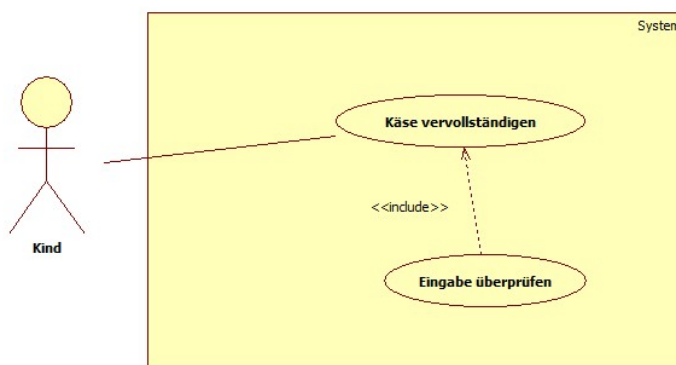


Abbildung 44 : (Eigene Darstellung) Use Case Diagramm Käse Spiel

Käse vervollständigen

Um einen Käse zu vervollständigen muss das Kind ein in der linken Spalte abgebildetes Stück Käse per Drag & Drop in die Lücke ziehen und so den Käse vervollständigen. Sobald das Stück auf das dafür vorgesehene Feld gezogen wurde, erscheint darunter die Bruchzahl des

Stücks. Mit Hilfe dieser Bruchzahl kann das Kind erkennen, dass z.B $19/20 + 1/20$, $20/20$ ergibt.

Eingabe überprüfen

Der Button mit dem die Eingabe überprüft werden kann erscheint erst nachdem das Kind ein Stück Käse auf das Feld gezogen hat. Mit Hilfe dieses Knopfs werden die Eingaben anschließend überprüft. Ist die Eingabe richtig so erhält das Kind 10 Bananen. Andernfalls verliert es 10 Bananen.

19.6.12 Kreisspiel

19.6.12.1 *Kurzbeschreibung und Übersicht*

Dieses Mini-Spiel befasst sich mit Kreisen. Den Kindern sollen auf eine spielerische Art vermittelt werden wie über wie viel Grad ein Kreis oder ein beispielsweise Halbkreis verfügt. Untenstehende Abbildung zeigt die Benutzeroberfläche des Mini-Spiels. Ziel des Spiels ist es, dass das Kind die abgebildete Zahl an Grad auf dem Kreis darstellt.

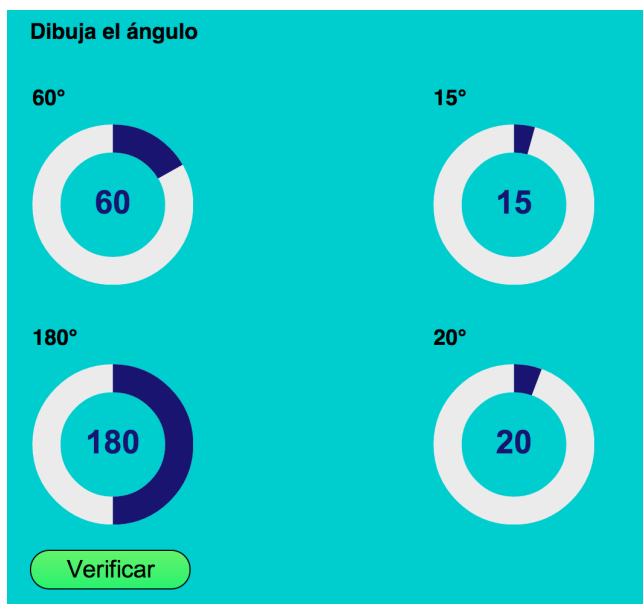


Abbildung 45: (Eigene Darstellung) Benutzeroberfläche Winkel und Kreise

19.6.12.2 Use Cases

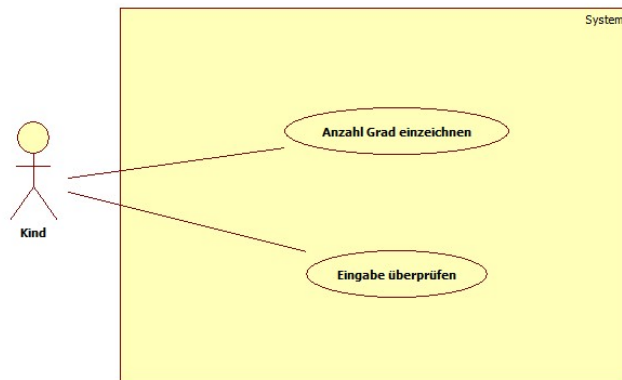


Abbildung 46 : (Eigene Darstellung) Use Case Winkel und Kreise

Anzahl Grad einzeichnen

Um die abgebildete Zahl auf dem Kreis anzugeben kann ein Punkt auf dem grauen Kreis angeklickt werden. Die Zahl in der Mitte gibt Auskunft über den aktuell selektierten Wert. Wurde der Kreis bereits einmal angeklickt, so kann der selektierte Wert durch Scrollen mit der Maus oder durch ein erneutes anklicken verfeinert werden.

Eingabe überprüfen

Mit Hilfe des „Verificar“ Buttons können die Eingaben überprüft werden. Sind alle Eingaben korrekt, so wird der Spieler mit 10 Bananen belohnt. Sind die Eingaben falsch, so wird der Bestand an Bananen um 10 reduziert.

19.6.13 Mathematiktablelle

19.6.13.1 Kurzbeschreibung und Übersicht

Dieses Spiel befasst sich mit dem Training der Grundrechenarten Subtraktion, Addition und Multiplikation. Im Feld der ersten Zeile und der ersten Spalte ist jeweils der Operator der trainiert wird angegeben. Das Kind muss nun die Tabelle vervollständigen und die Zahl der jeweiligen Spalte mit der Zahl der jeweiligen Zeile gemäß angegebenem Operator berechnen.

Completa la tabla

-	4	3	2	5
8	4	5	6	3
9	5	6	7	4
7	3	4	5	2

Verificar

Abbildung 47 : (Eigene Darstellung) Benutzeroberfläche Mathtable

19.6.13.2 Use Cases

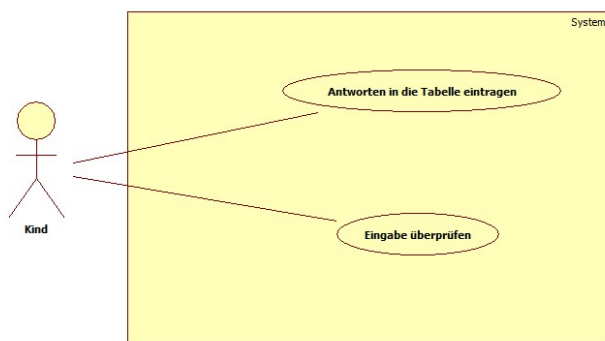


Abbildung 48 : (Eigene Darstellung) Use Case Diagramm Math Table

Antworten in die Tabelle eintragen

Um die Antworten in die Tabelle einzutragen genügt ein einfacher Klick auf das Eingabefeld. Die Zahl kann anschließend ins Eingabefeld mit Hilfe der Zahlentasten eingetragen werden.

Eingabe prüfen

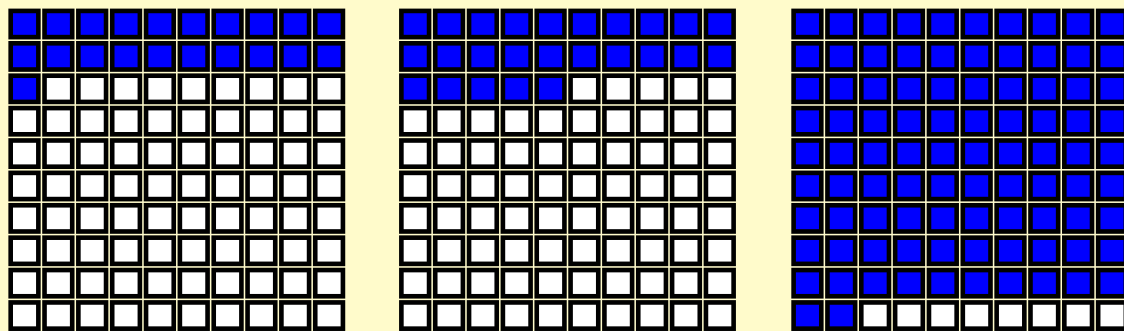
Auch hier werden die Eingaben mit Hilfe des „Verificar“ Buttons geprüft. Sind alle Eingaben korrekt so erhält der Spieler 10 Bananen. Andernfalls wird der Bananenbestand um 10 reduziert.

19.6.14 Gitteradditionsspiel

19.6.14.1 Kurzbeschreibung und Übersicht

Dieses Spiel soll die Additionskenntnisse der Kinder verbessern. Ziel des Spiels eine Rechnung mit einer Unbekannten zu komplettieren. Das Resultat der Rechnung beläuft sich immer auf Hundert.

Completa la tarea haciendo clic en la casilla correcta

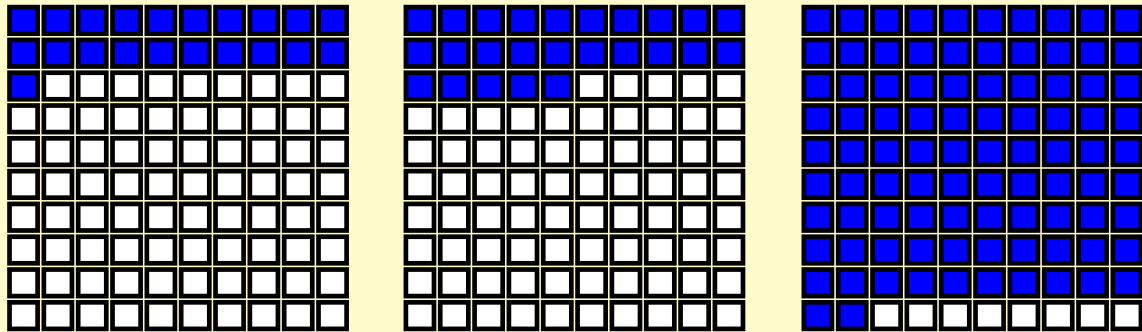


$$92 + _ _ = 100$$

Abbildung 49 : (Eigene Darstellung) Benutzeroberfläche Gitteradditionsspiel vor Auswahl

Das Kind muss den richtigen der drei eingefärbten Blöcke auswählen. Nach Auswahl eines Blockes wird die Rechnung komplettiert und der „Continuar“ Knopf erscheint.

Completa la tarea haciendo clic en la casilla correcta



$$92+8=100$$

Verificar

Abbildung 50 : (Eigene Darstellung) Benutzeroberfläche Gitterraditionsspiel nach Auswahl

19.6.14.2 Use Cases

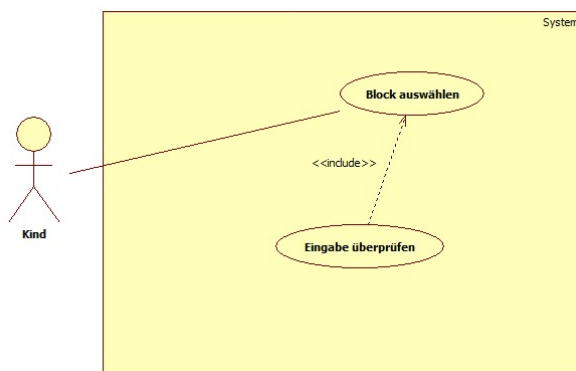


Abbildung 51 : (Eigene Darstellung) Use Case Gitterraditionsspiel

Block auswählen

Ein Block kann ganz einfach durch einen einfachen Klick ausgewählt werden. Bei der Auswahl eines Blockes wird die Rechnung automatisch vervollständigt.

Eingabe überprüfen

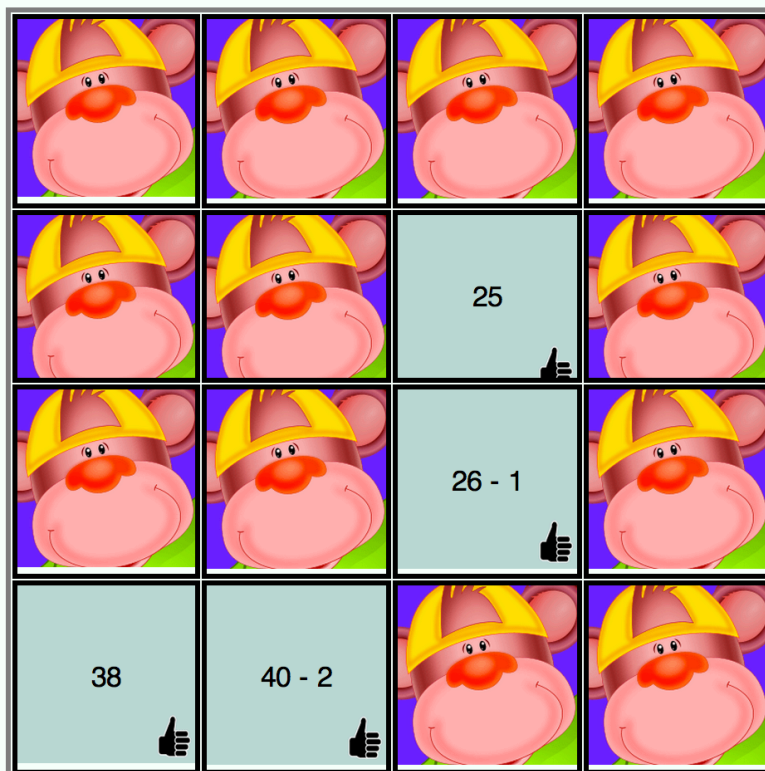
Wie aus dem UML Diagramm erkennbar ist die Überprüfung der Eingabe erst nach Auswahl eines Blockes möglich. Auch hier werden bei Korrekter Beantwortung der Fragenstellung 10 Bananen vergeben und ansonsten 10 Bananen abgezogen.

19.6.15 Memory-Spiel

19.6.15.1 Kurzbeschreibung und Übersicht

Wie der Titel des Spiels bereits aussagt handelt es sich bei diesem Mini-Spiel um ein Memory-Spiel. Im Gegensatz zum herkömmlichen Memory-Spiel gilt es nicht zwei gleiche Bilder, sondern zwei gleiche Zahlen oder Resultate zu finden. Je nach Level werden auf den Rückseiten der Memory-Karten Aufgaben mit Additionsaufgaben, Subtraktionsaufgaben, Multiplikationsaufgaben oder Divisionsaufgaben gestellt. Ziel ist es das Memory mit möglichst wenigen Zügen zu beenden.

Encuentra las parejas



Intiendos: 8

≤ 20 Intentos = 10 Plátanos
 ≤ 40 Intentos = 0 Plátanos
 > 40 Intentos = -10 Plátanos

Abbildung 52 : (Eigene Darstellung) Benutzeroberfläche Memory-Spiel

19.6.15.2 Use Cases

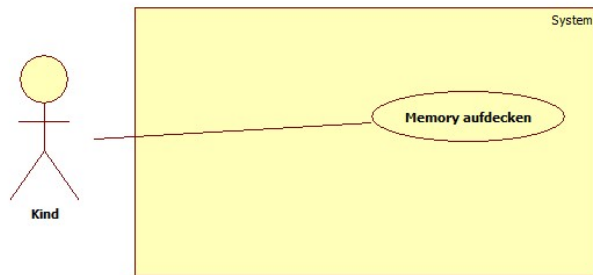


Abbildung 53 : (Eigene Darstellung) Use Case Memory-Spiel

Memory aufdecken

Bei diesem Mini-Spiel gibt es nur einen Use Case. Ein Memory kann durch einen einfachen Klick aufgedeckt werden. Nachdem die zweite Memory-Karte aufgedeckt wurde, werden die Aufgaben und Resultate der aufgedeckten Karten automatisch auf ihre Gleichheit überprüft. So können beispielsweise eine Karte mit der Aufschrift $3 + 2$ und eine Karte mit der Aufschrift 5, zwei Karten mit der Aufschrift 5 oder beispielsweise eine Karte mit der Aufschrift $2 + 3$ und eine Karte mit der Aufschrift $4 + 1$ aufgedeckt werden. Sind die Resultate der Karten gleich, so werden die Karten grau eingefärbt und können von nun an nicht mehr angeklickt werden. Die Bananen werden anschließend unter Berücksichtigung der benötigten Anzahl Versuche vergeben. Bleibt der Spieler unter 20 Versuchen so erhält er 10 Bananen. Bleibt der Spieler unter 40 so erhält er keine Banane aber verliert auch keine Bananen. Bei mehr als 40 Versuchen verliert der Spieler 10 Bananen.

19.6.16 Schloss Mini-Spiel

19.6.16.1 Kurzbeschreibung und Übersicht

Mit Hilfe dieses Mini-Spiels werden wiederum die vier Grundrechenarten trainiert. Ziel des Spiels ist es mit Kanonen auf ein Schloss zu feuern und dieses einzunehmen.

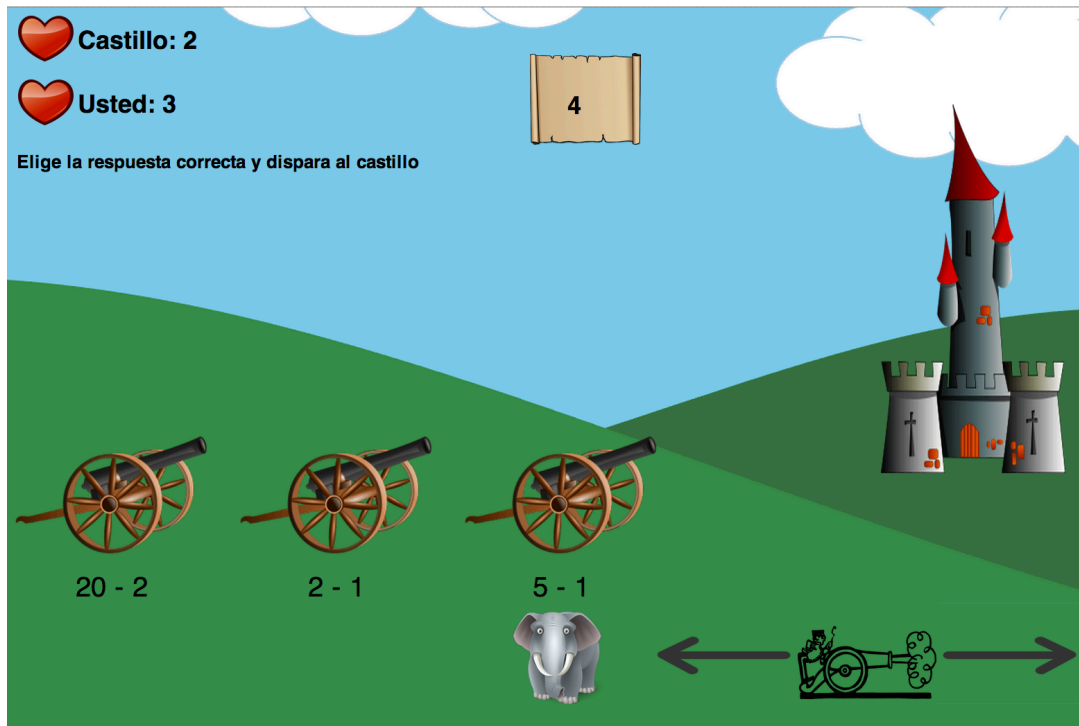


Abbildung 54 : (Eigene Darstellung) Benutzeroberfläche Schloss Mini-Spiel

Der Spieler muss die Spielfigur unter der Kanone mit der passenden Rechnung platzieren. Die Rechnung ist dann passend wenn ihr Resultat mit der auf der Papierrolle abgebildeten Zahl übereinstimmt. Hat der Spieler die Figur an der richtigen Stelle platziert kann er auf das Schloss feuern. Stimmt die Rechnung allerdings nicht mit der oben angegebenen Zahl überein so feuert das Schloss auf die Spielfigur. Sobald ein Treffer stattfindet wird die Anzahl der angegebenen Leben oben rechts um eins reduziert.

19.6.16.2 Use Cases

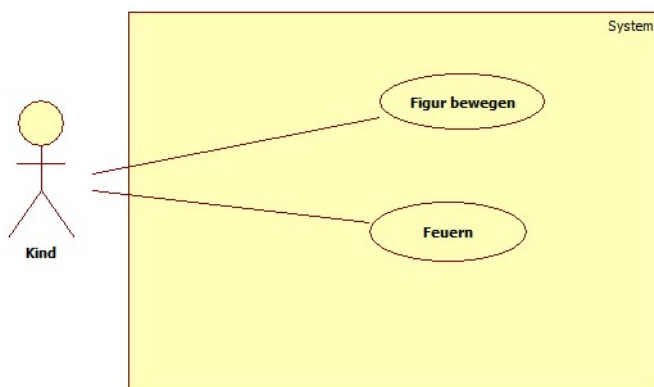


Abbildung 55 : (Eigene Darstellung) Use Case Castlegame

Figur bewegen

Um die Spielfigur unter einer Kanone zu platzieren, können die Pfeiltastenknöpfe am unteren Rand des Fensters benutzt werden.

Feuern

Die Kanone kann durch einen Klick auf das Bild des Mannes eine Kanone abfeuern. Wurde die Figur zuvor unter dem richtigen Resultat platziert so wird die Kanone auf da Schloss gefeuert und die Anzahl der Leben des Schlosses um 1 reduziert. Wurde der Spieler unter einer Rechenaufgabe platziert die nicht mit dem abgebildeten Resultat übereinstimmt so feuert das Schloss auf den Spieler. Die Anzahl Leben des Spielers wird anschließend dekrementiert. Verliert das Schloss zuerst seine drei Leben so gewinnt der Spieler und erhält zehn Bananen. Verliert der Spieler, wird also seine Anzahl Leben auf Null reduziert, so verliert er auch zehn Bananen.

19.6.17 Teil Spiel

19.6.17.1 Kurzbeschreibung und Übersicht

Bei diesem Spiel werden die Divisionskenntnisse der Kinder verbessert. Ziel ist es die Äpfel gleichmäßig auf Tellern zu platzieren so dass auf allen Tellern gleich viele Äpfel liegen.

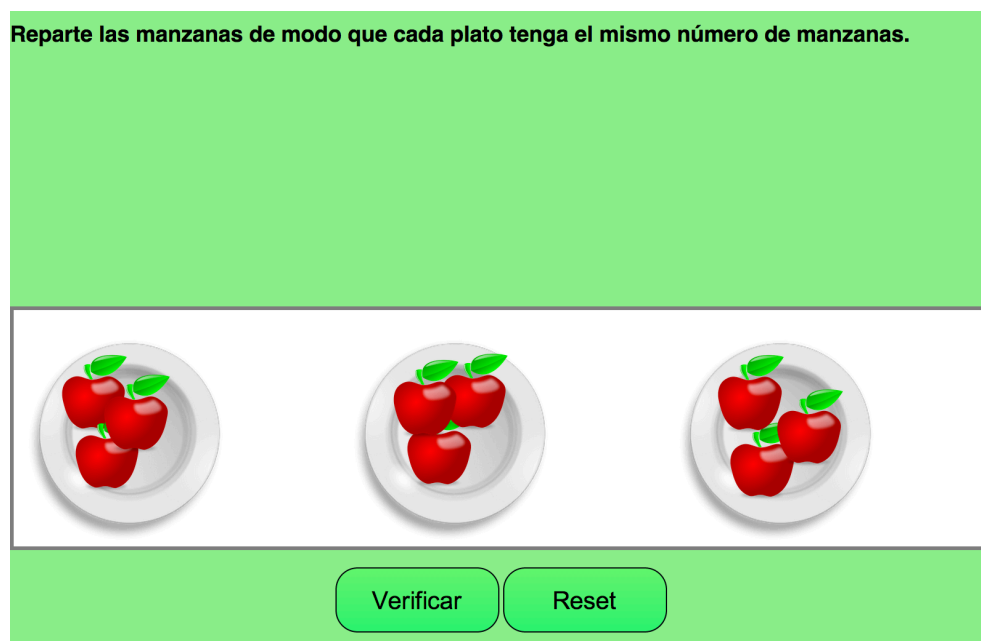


Abbildung 56 : (Eigene Darstellung) Benutzeroberfläche Teil Spiel

19.6.17.2 Use Cases

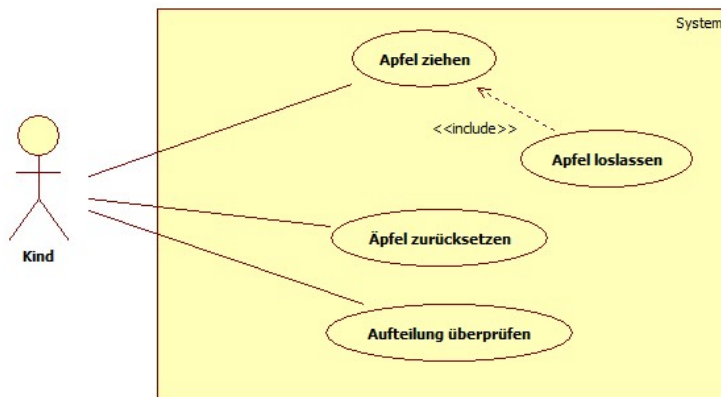


Abbildung 57 : (Eigene Darstellung) Use Case Teil Spiel

Apfel ziehen & Apfel loslassen

Der Apfel kann einfach per Drag & Drop gezogen werden. Ein Apfel muss auf einem Teller losgelassen werden ansonsten wird er automatisch auf die Ausgangsposition zurückgelegt.

Äpfel zurücksetzen

Mit Hilfe des „Espalda“ Buttons können alle Äpfel auf die Ausgangsposition zurückgesetzt werden.

Aufteilung überprüfen

Mit Hilfe des „Verificar“ Buttons kann überprüft werden ob die Äpfel richtig verteilt wurden. Wurden die Äpfel richtig platziert so erhält der Spieler zehn Bananen plus. Wurden die Äpfel allerdings falsch platziert verliert der Spieler zehn Bananen.

19.7 Sonderfälle

Während einer Spielrunde können zwei Sonderfälle eintreten. Entweder begegnet man dem Tiger oder dem Affen. Diese zwei Fälle werden als Sonderfall bezeichnet, da sie beim Spielen einer ganzen Runde auf jeden Fall einmal eintreten. Damit man beispielsweise durch Würfelpech die Bananenauslieferung verpasst, wurde das Spiel so implementiert das bei goldenen und schwarzen Feldern auf jeden Fall angehalten wird.

19.7.1 Goldenes Feld – Besuch beim Affen

Beim Affen hat der Spieler die Möglichkeit seine zuvor erspielten Bananen gegen Geld einzutauschen. Mit Hilfe dieses Geldes können anschließend verschiedene Levels freigeschaltet werden. Untenstehendes Aktivitätsdiagramm zeigt den Ablauf des Besuchs beim Affen.

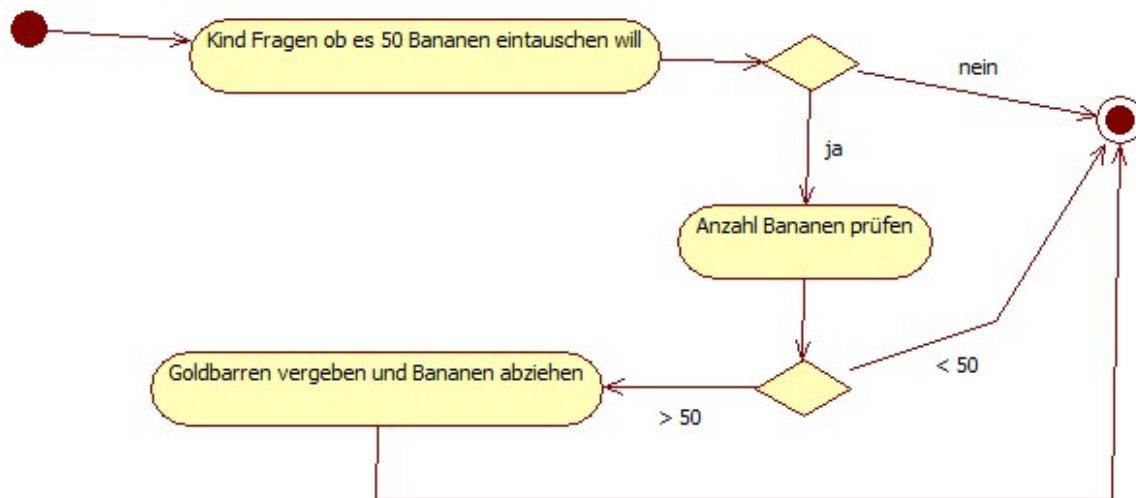


Abbildung 58 : (Eigene Darstellung) Aktivitätsdiagramm – Bananenabgabe

Zu Beginn erscheint das Eingangsbild der Abgabe. Das Kind hat anschließend die freie Wahl die Bananen einzutauschen oder nicht. Wählt das Kind den Tausch, so werden die Anzahl Bananen geprüft. Verfügt das Kind über genügend Bananen, so wird dem Kind eine Goldbarre vergeben. Andernfalls ist ein Tausch nicht möglich. Untenstehendes Bild zeigt ein Screenshot der Bananenabgabe des dritten Levels.



Abbildung 59 : (Eigene Darstellung) Abgabe der Bananen Level 3

19.7.2 Schwarzes Feld - Tiger

Der Tiger symbolisiert das Böse. Gelangt man auf ein schwarzes Feld so spielt man in einem direkten Duell gegen den Tiger. Dieses Duell stellt eine Art Prüfung dar. Abhängig vom Level werden hier Additions-, Subtraktions-, Multiplikations- und Divisionsaufgaben gestellt. Der Tiger wird durch eine künstliche Intelligenz repräsentiert. Dieses Duell kann drei verschiedene Ausgänge nehmen.

1. Sieg des Spielers
2. Unentschieden
3. Sieg des Tigers

Beim Sieg des Spielers verliert der Spieler keine Bananen. Allerdings gewinnt er auch keine Bananen. Bei einem Unentschieden verliert der Spieler 10 Bananen. Beim Sieg des Tigers verliert der Spieler 50 Bananen. Dies mag hart erscheinen, jedoch soll dadurch verhindert werden, dass der Spieler das nächste Level erreicht obwohl er den Stoff noch nicht beherrscht.

19.7.2.1 Spielablauf

Bei diesem Spiel werden wiederum die 4 Grundrechenarten trainiert. Der Spieler spielt hierbei gegen eine künstliche Intelligenz, den Tiger. Das Spiel bildet eine Art Quizduell bei dem es darum geht mathematische Aufgaben richtig zu beantworten. Derjenige Spieler der zuerst 6 Aufgaben richtig beantwortet gewinnt das Spiel.

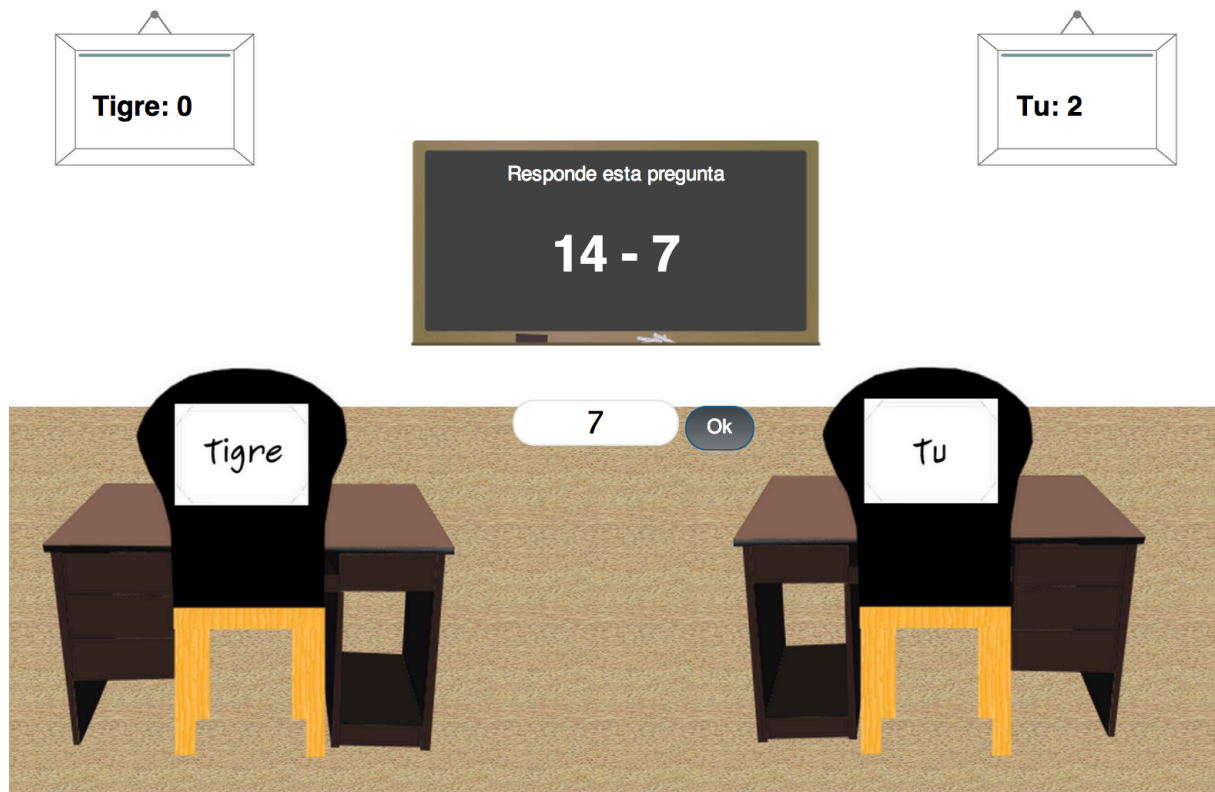


Abbildung 60 : (Eigene Darstellung) Quizgame Eingabe

Zuerst wird die Antwort eingegeben. Die Eingabe wird dann mit Ok bestätigt. Im Anschluss wird sowohl die Antwort des Tigers als auch die Antwort des Spielers angezeigt. Bei einer richtigen Antwort wird die Punktezahl des Spielers der richtig Antwortet um 1 erhöht.

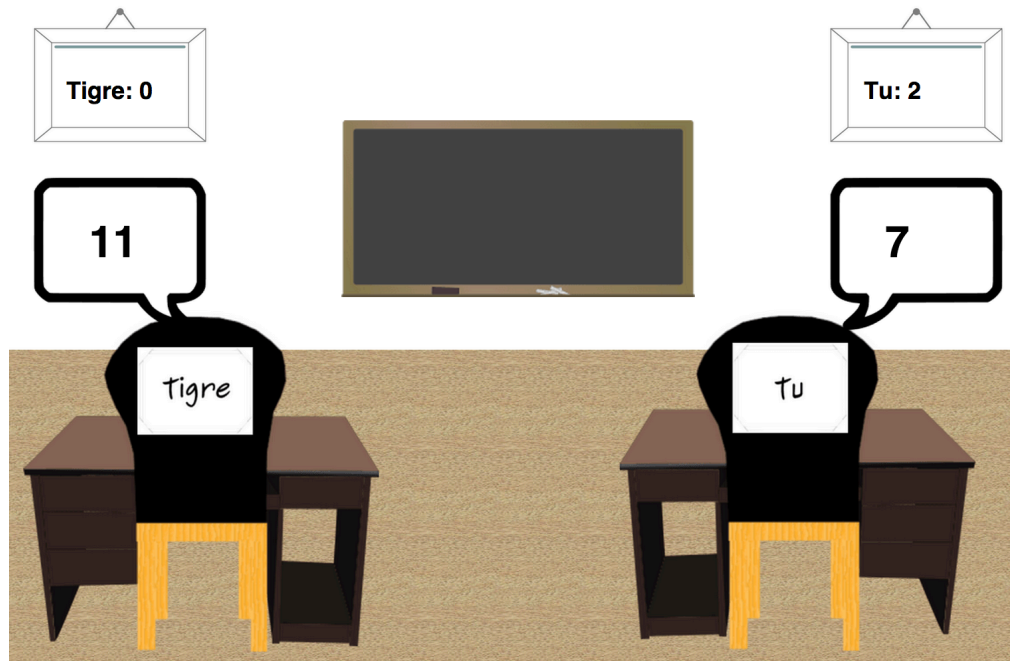


Abbildung 61 : (Eigene Darstellung) Quizgame nach Eingabe

19.7.2.2 Use Cases

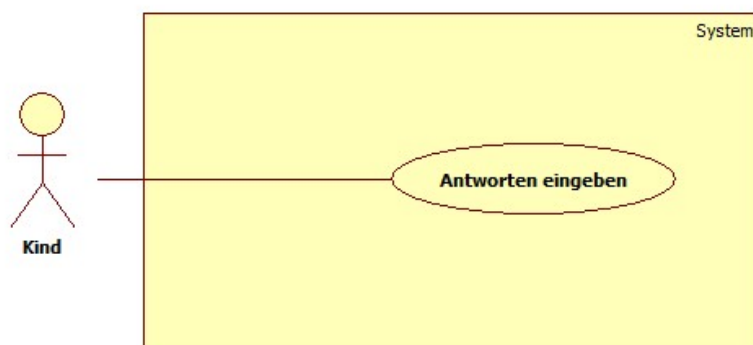


Abbildung 62 : (Eigene Darstellung) Use Case Quizgame

Antworten eingeben

Die Antwort kann ins Eingabefeld eingegeben werden. Mit Ok wird die Antwort bestätigt. Ist die Antwort richtig, so wird die Punktezahl des Spielers um 1 erhöht. Für das Quizspiel gibt es 3 verschiedene Ausgänge. Sieg des Tigers, Unentschieden oder Sieg des Spielers. Da es sich um das Spiel gegen den Tiger welcher der Bösewicht symbolisiert handelt, kann der Spieler nur verlieren. Gewinnt also der Tiger so verliert der Spieler dreißig Bananen, Endet das Duell Unentschieden so verliert der Spieler zehn Bananen. Alleine bei einem Sieg des Spielers verliert dieser keine Bananen.

Technische Dokumentation

In diesem Teil wird ein kleiner Auszug des Quellcodes erläutert.
Darüber hinaus werden verwendete Bibliotheken und
sonstige technische Inhalte vorgestellt.

20 Vorbereitung des Systems

Die OLPC Laptops benutzen Sugar als grafische Benutzeroberfläche. Sugar ist eine speziell angefertigte graphische Benutzeroberfläche für Kinder. Ziel ist es, dass sie den Kindern den Umgang mit dem Computer durch selbsterklärende Symbole erleichtert. Durch verschiedene Sugar Aktivitäten können die Kinder lernen. Da Sugar auf einem Linux Fedora basiert habe ich eine neue Virtuelle Maschine mit einem Linux Fedora 19 – „Schrödingers Cat“ aufgesetzt. Eine Sugar Web Aktivität kann sowohl mit Sugarizer als auch mit osbuild implementiert werden. Für meine Arbeit habe ich mich für die osbuild Variante entschieden. Osbuild ist die Sugar build Umgebung. Osbuild stellt eine volle Sugar Umgebung zur Verfügung. Sugarizer hingegen simuliert nur eine Sugar Umgebung und braucht für die Entwicklung lediglich einen herkömmlichen Browser. (wikipedia.org, 2014), (Developer Sugarlabs)

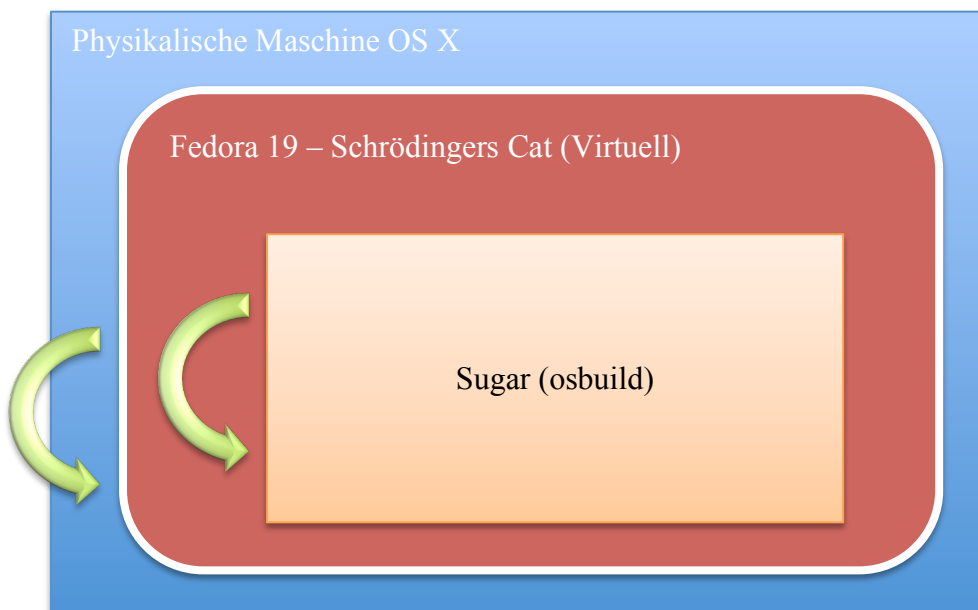


Abbildung 63 : (Eigene Darstellung) Systemarchitektur

Sugar besteht aus mehreren Modulen und hängt oft von in Linux nicht mitgelieferten Klassenbibliotheken ab. Sugarlabs stellt eine Reihe an Skripten bereit, welche builds und andere Entwicklungsaufgaben automatisieren. Um Sugar zu installieren sind folgende Schritte auszuführen. (Sugarlabs)

1. Um die neuste Sugar Version zu erlangen muss zuerst Git installiert werden. Mit dem Befehl:

```
yum install curl-devel expat-devel gettext-devel \ openssl-devel zlib-devel
```

können zunächst alle Abhängigkeiten welche für Git benötigt werden installiert werden. Um diese Abhängigkeiten zu installieren muss man über root Berechtigungen verfügen. (Sugarlabs)

2. Nachdem die benötigten Abhängigkeiten installiert sind, muss in einem Linux Fedora noch folgender Befehl abgesetzt werden.

```
yum install git
```

(Sugarlabs)

3. Nachdem Git erfolgreich installiert wurde muss man zunächst das sugar-build Git repository klonen.

```
Git clone git://github.com/sugarlabs/sugar-build.git
```

(Sugarlabs)

4. Anschließend muss man ins Hauptverzeichnis wechseln und dort einen „pull“ ausführen um den Source Code zu erlangen. Hierbei ist es wichtig, dass man zuvor mit dem Befehl su den Benutzer wechselt da der Pull nicht mit dem root ausgeführt werden soll.

```
cd sugar-build  
./osbuild pull
```

(Sugarlabs)

5. Nach Beendigung des pull Befehls muss man eine „build“ ausführen.

```
./build
```

(Sugarlabs)

6. Wenn all diese Schritte erfolgreich ausgeführt wurden kann man Sugar mit dem „run“ Befehl starten.

```
run
```

Nach dem run Befehl startet Sugar automatisch. Man befindet sich anschließend in einem vollen Sugar Umfeld. (Sugarlabs)

Web Aktivität

Mit untenstehendem Befehl kann eine neue Webaktivität erstellt werden.

```
./osbuild shell  
volo create meineaktivität ./sugar-web-template
```

Die erstellte Aktivität weist folgende Struktur auf.

- meineaktivität
 - activity
 - activity.info
 - activity-icon.svg
 - HTML Seiten
 - CSS
 - Alle CSS Dateien
 - js
 - activity.js
 - loader.js
 - lib
 - package.json

- setup.py

Quelle: (Sugarlabs)

Der Ordner Activity enthält alle Informationen wie Name, Icon und ID der Aktivität. Im js Ordner befinden sich sämtliche Java Script Dateien und somit die ganze Logik der Applikation. Im CSS Ordner werden alle CSS Dateien welche zur Gestaltung der Webseite gebraucht werden abgelegt. Im Ordner lib sind alle benötigten Klassenbibliotheken abgelegt. Die Datei package.json beinhaltet alle Informationen betreffen Abhängigkeiten. Die Datei setup.py erlaubt es die Datei zu installieren oder ein installierbares Bundle zu erstellen. (Sugarlabs)

21 Verwendete Bibliotheken

Neben den gängigen Webtechnologien wie HTML, CSS und Javascript habe ich zusätzlich einige Javascript Bibliotheken verwendet. Diese verschiedenen Bibliotheken werde in diesem Kapitel kurz vorgestellt. Zudem werden zu manchen Bibliotheken kurze Beispiele aus meiner Applikation vorgestellt.

21.1 D3 – Data-Driven Documents

D3 ist eine Javascript Bibliothek die die Manipulation von Dokumenten basierend auf Daten erlaubt. D3 beinhaltet vielfältige Visualisierungen und Document Object Model (DOM) Manipulationen. In meiner Bachelorarbeit wurde die D3 Bibliothek für jegliche Zeichnungen von SVG Elementen benutzt. Besonderen Einsatz fand diese Bibliothek bei der Erstellung der Spielfelder. (Github.com, 2014)

Die Daten des jeweiligen Feldes wurden als JavaScript Object Notation (JSON) Daten gespeichert. Untenstehende Abbildung zeigt ein Ausschnitt der Daten der Felder von Level1.


```

var circleData;

//Circle Data Set
var level1 = [
  { "cx": 50, "cy": 770, "radius": 30, "color": "snow", "fontcolor": "black", "number": 1 },
  { "cx": 120, "cy": 700, "radius": 30, "color": "blue", "fontcolor": "white", "number": 2 },
  { "cx": 190, "cy": 700, "radius": 30, "color": "green", "fontcolor": "white", "number": 3 },
  { "cx": 260, "cy": 700, "radius": 30, "color": "orange", "fontcolor": "black", "number": 4 },
  { "cx": 330, "cy": 720, "radius": 30, "color": "pink", "fontcolor": "black", "number": 5 },
  { "cx": 400, "cy": 720, "radius": 30, "color": "purple", "fontcolor": "black", "number": 6 },
  { "cx": 470, "cy": 720, "radius": 30, "color": "pink", "fontcolor": "black", "number": 7 },
  { "cx": 540, "cy": 720, "radius": 30, "color": "silver", "fontcolor": "black", "number": 8 },
  { "cx": 610, "cy": 720, "radius": 30, "color": "lavender", "fontcolor": "black", "number": 9 },
  { "cx": 680, "cy": 720, "radius": 30, "color": "red", "fontcolor": "white", "number": 10 },
  { "cx": 750, "cy": 700, "radius": 30, "color": "green", "fontcolor": "black", "number": 11 },
  { "cx": 820, "cy": 680, "radius": 30, "color": "silver", "fontcolor": "black", "number": 12 },
  { "cx": 890, "cy": 660, "radius": 30, "color": "orange", "fontcolor": "black", "number": 13 },
  { "cx": 880, "cy": 590, "radius": 30, "color": "turquoise", "fontcolor": "white", "number": 14 },
  { "cx": 810, "cy": 580, "radius": 30, "color": "greenyellow", "fontcolor": "black", "number": 15 },
  { "cx": 740, "cy": 580, "radius": 30, "color": "mediumaquamarine", "fontcolor": "black", "number": 16 },
  { "cx": 670, "cy": 580, "radius": 30, "color": "green", "fontcolor": "black", "number": 17 },
  { "cx": 600, "cy": 580, "radius": 30, "color": "red", "fontcolor": "white", "number": 18 },
  { "cx": 530, "cy": 580, "radius": 30, "color": "mediumaquamarine", "fontcolor": "black", "number": 19 },
  { "cx": 460, "cy": 580, "radius": 30, "color": "silver", "fontcolor": "black", "number": 20 },
  { "cx": 390, "cy": 580, "radius": 30, "color": "midnightblue", "fontcolor": "white", "number": 21 },
  { "cx": 320, "cy": 580, "radius": 30, "color": "silver", "fontcolor": "black", "number": 22 },
  { "cx": 320, "cy": 510, "radius": 30, "color": "blue", "fontcolor": "white", "number": 23 },
  { "cx": 390, "cy": 500, "radius": 30, "color": "turquoise", "fontcolor": "black", "number": 24 },
  { "cx": 460, "cy": 500, "radius": 30, "color": "green", "fontcolor": "white", "number": 25 },
  { "cx": 520, "cy": 500, "radius": 30, "color": "salmon", "fontcolor": "black", "number": 26 },
  { "cx": 590, "cy": 500, "radius": 30, "color": "red", "fontcolor": "white", "number": 27 },

```

Abbildung 64 : (Eigene Darstellung) Felddaten Level 1

Für jedes Feld werden sowohl die x Koordinate – den vertikalen Abstand zum rechten Rand sowie die y Koordinate – den horizontalen Abstand zum oberen Rand in Pixel gespeichert. Weiter wird für jedes Feld der Radius angegeben. Hinzu kommt eine Hintergrundfarbe sowie eine Vordergrundfarbe die die Farbe der Nummer des Feldes bestimmt. Als letztes wird die angezeigte Nummer des Feldes gespeichert. Bei jeder Bewegung des Spielers werden die Koordinaten des aktuellen Feldes gelesen und die Spielfigur anschließend anhand dieser Koordinaten neu gezeichnet. Auch bei der Feldüberprüfung wird die Farbe anhand dieser Daten gelesen.

```
function drawgameboard(){

//Add circles to the svgContainer
var circles = svgContainer.selectAll("circle")
    .data(circleData)
    .enter()
    .append("circle");

//Add the circle attributes
var circleAttributes = circles
    .attr("cx", function (d) { return d.cx; })
    .attr("cy", function (d) { return d.cy; })
    .attr("r", function (d) { return d.radius; })
    .style("fill", function (d) { return d.color; }).attr("stroke","black");

//Add the SVG Text Element to the svgContainer
var text = svgContainer.selectAll("text")
    .data(circleData)
    .enter()
    .append("text");

//Add SVG Text Element Attributes
var textLabels = text
    .attr("x", function(d) { return d.cx-5; })
    .attr("y", function(d) { return d.cy+5; })
    .text(function(d) { return d.number; })
    .attr("font-family", "Chalkboard")
    .attr("font-size", "20px")
    .attr("fill", function(d) { return d.fontcolor; });
```

Abbildung 65 : (Eigene Darstellung) Elemente auf dem Spielbrett zeichnen

Mit Hilfe der D3 Bibliothek werden nun die Elemente auf den Startbildschirm gezeichnet. Zuerst werden die Kreise Mit Hilfe der zuvor angegebenen Daten gezeichnet. Danach werden die zuvor angegebenen Kreisattribute einzeln herausgelesen und vergeben. Nachdem die Kreise gezeichnet wurden werden noch die Textlables hinzugefügt.

21.2 JQuery

JQuery ist eine schnelle elegante Javascript Bibliothek die über viele verschiedene Funktionen verfügt. JQuery vereinfacht die Manipulation des HTML Dokuments. Weiter unterstützt JQuery verschiedenste Ein- und Ausblendungseffekte. In meiner Arbeit wurde JQuery besonders für das Verstecken einiger Elemente, Einblende Effekte, Ausblende Effekte, Animationen oder Drag & Drop verwendet. Neben der reinen JQuery Bibliothek wurden weiter verschiedene JQuery Plugins verwendet. (jquery.com, 2014)

JQuery Knob

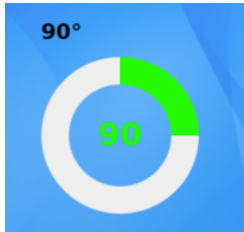


Abbildung 66 : (Eigene Darstellung) JQuery Knob Kreis

Das JQuery Knob Plugin wurde für die Erstellung des Kreisspiels verwendet. Mit Hilfe dieses Plugins lassen sich input Elemente als Kreise darstellen. Auf diesen Kreisen kann anschließend ein Regler hin und her bewegt werden. Beim Bewegen des Reglers ändert sich der in der Mitte des Kreises abgebildete Wert. Untenstehender Code zeigt wie ich dieses Plugin in meiner Arbeit verwendet habe.

Im ersten Schritt habe ich das Plugin eingebunden. Bei der Einbindung eines JQuery Plugins gilt es zu beachten, dass das Plugin nach der JQuery Bibliothek eingebunden wird. Allerdings muss es auch vor den vom Benutzer erstellten Javascript Dateien eingebunden werden. Danach wird ein input Element erstellt und wie gewünscht platziert.

```
<td>
    <h1 id="circlehl_1">180°</h1>
    <input id="kendocircle1" type="text" data-min="0" data-max="360" value="0" data-fgColor="lime" class="dial">
</td>
```

Abbildung 67 : (Eigene Darstellung) Codebeispiel Kendo

Das Input Element verfügt über folgende Attribute. Den Minimumwert, den Maximumwert, die Farbe etc... Den Rest erledigt dabei das Plugin. Die Id dient dazu die Werte des Elements zu lesen und anschließend in der Logik zu verwenden.

22 Single-page Web Applikation

Bereits bei den ersten Testprogrammen konnte ich feststellen, dass bei einer Sugar Web Activity nur mit einer einzigen HTML Seite gearbeitet werden kann. Um den Wechsel zwischen den Mini-Spielen und der Hauptaktivität zu realisieren habe ich hierzu die JQuery hide() funktion benützt. Dies bedeutet, dass zur Laufzeit lediglich die benötigten Elemente angezeigt werden. Die Interaktion dabei läuft über das zentrale Script, das mainscript.js. Untenstehendes Sequenzdiagramm verdeutlicht die Interaktion zwischen dem User, der HTML Seite, dem Mainscript und der Spielfigur.

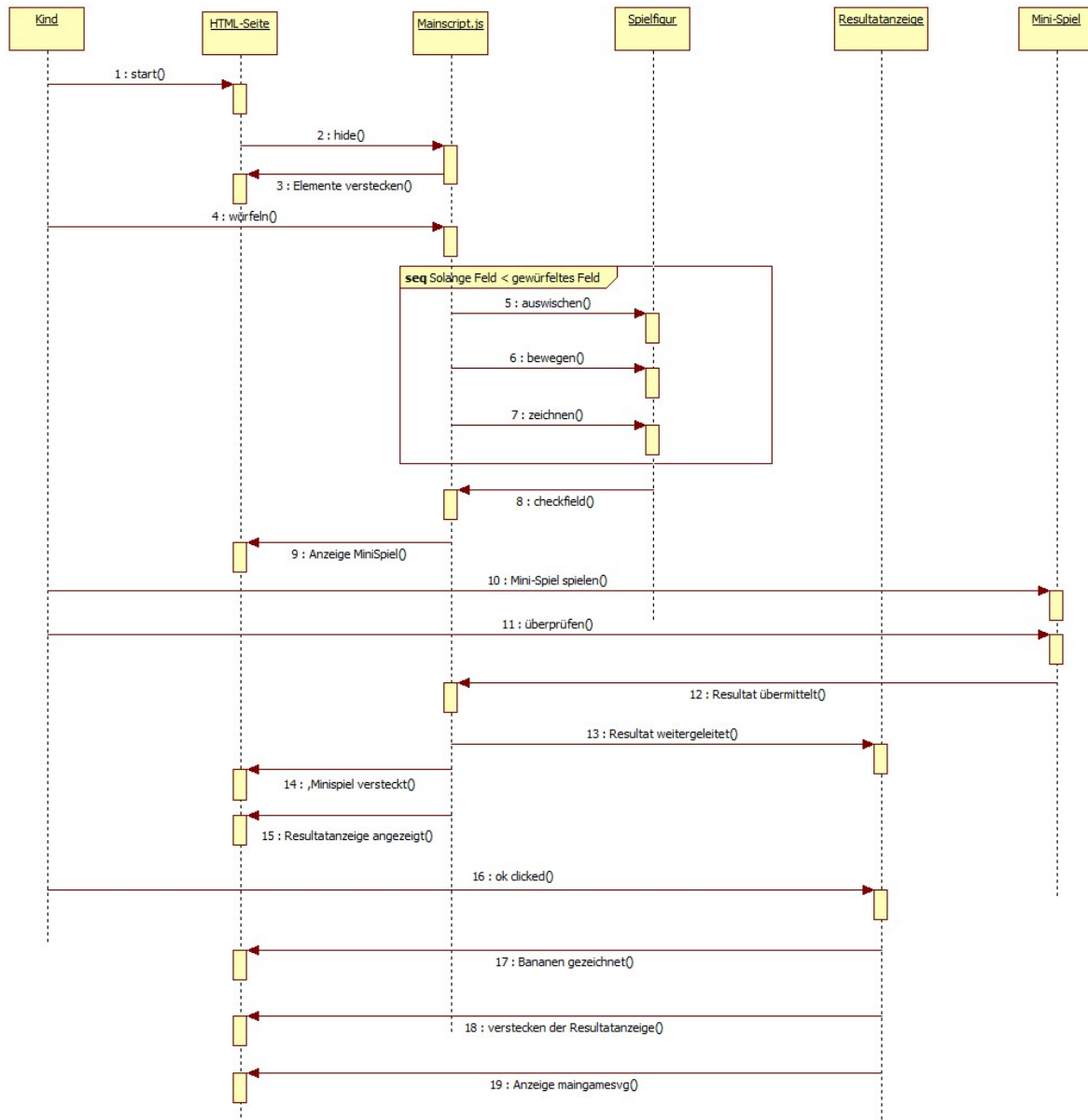


Abbildung 68 : (Eigene Darstellung) Sequenzdiagramm einzelne HTML Seite

Beim Start des Spiels wird die Methode `hide()` des `mainscripts` aufgerufen. Diese Methode versteckt alle nicht benötigten Elemente. Dadurch wird einzig und allein das benötigte Element angezeigt.

```
function hide(){
$( "#shopdiv" ).hide();
$( "#maingamesvg" ).hide();
$( "#krokodiv" ).hide();
$( "#formgame" ).hide();
$( "#endresultgame" ).hide();
$( "#whatnextgame" ).hide();
$( "#cheese" ).hide();
$( "#tengame" ).hide();
$( "#divisiongame" ).hide();
$( "#bananadelivery" ).hide();
$( "#slidergame" ).hide();
$( "#wallgame" ).hide();
$( "#endofround" ).hide();
$( "#sourcediv" ).hide();
$( "#circlegame" ).hide();
$( "#multiplikationtablediv" ).hide();
$( "#gridaddition" ).hide();
$( "#sharinggame" ).hide();
$( "#memorygame" ).hide();
$( "#castlegame" ).hide();
$( "#bossquiz" ).hide();
$( "#tigerdiv" ).hide();
$( "#progress_buttondiv" ).hide();
$( "#characterselection" ).hide();
$( "#newleveldiv" ).hide();

}

```

Abbildung 69 : (Eigene Darstellung) Methode Hide()

Danach würfelt der Spieler. Der Würfelwurf löst anschließend die Bewegung des Spielers aus. Dazu entfernt das mainscript die Spielfigur vom Spielbrett, berechnet anschließend die neue Position und zeichnet dann die Spielfigur wiederum auf das Brett.

```
function drawlion($cx, $cy){

//Play the move sound
sound_blop.play();

svgContainer.append("svg:image")
    .attr("xlink:href", filename).attr("id", "Lion")
    .attr("width", 100)
    .attr("height", 100).attr("x", $cx-50).attr("y", $cy-90);
}

function deletelion(){

var lion = document.getElementById("Lion");
lion.parentNode.removeChild(lion);
}

```

Abbildung 70 : (Eigene Darstellung) Methode Draw und Delete

Die abgebildete Methode drawlion erhält die Position an der die Spielfigur gezeichnet werden soll. Diese Koordinaten stammen von den JSON Daten. Die Funktion deletelion ist für das Entfernen der Spielfigur vom Spielbrett verantwortlich.

Nachdem die Spielfigur um die Anzahl der gewürfelten Augenzahl bewegt worden ist, so wird die Farbe des aktuellen Feldes überprüft. Die Farbe des Feldes stammt dabei wiederum

aus den JSON Daten die zuvor vorgestellt wurden. Die Überprüfung findet dabei im mainscript statt und sieht wie folgt aus:

```
//Say which method needs to be called. Corresponding to the color the caracer is standing on.
switch(currentcolor){
case "green":
    green();
    break;
case "red":
    red();
    break;
case "blue":
    //Blue = Shop Game
    //Hide the main game
    $("#maingamesvg").hide();
    sound_shopbackground.play(); //Play the background music for the shopgame
    $("#shopdiv").show("Clip");
    createexercise();
    break;
case "yellow":
    //Yellow = Krokogame
    $("#maingamesvg").hide();
    sound_background_krokogame.play();
    $("#krokodiv").show();
    break;
case "orange":
    //Orange == formgame
    $("#maingamesvg").hide();
    sound_background1.play();
    createformsgame();
    $("#formgame").show();
    break;
```

Abbildung 71 : (Eigene Darstellung) Method Checkfield im Mainscript

Sobald die Farbe eines Feldes zutrifft wird eine Funktion ausgeführt. Bei allen Feldern welche ein Minispiel hervorrufen wird zuerst das div Element mit der id „maingamesvg“ – das Spielbrett, versteckt. Im Anschluss wird der Sound des Spiels abgespielt und je nach Bedarf das Spiel initialisiert. Danach wird das div Element mit der entsprechenden Id des Minispiels angezeigt und das Minispiel kann beginnen.

Im nächsten Schritt spielt der Spieler das Mini-Spiel und überprüft seine Aufgabe. Dieses Resultat wird anschließend ans Mainscript weitergeleitet. Das Mainscript leitet dieses Resultat an die Resultatanzeige weiter und initialisiert diese. Danach versteckt das Mainscript das Minispiel und zeigt die Resultatanzeige an.



Abbildung 72 : (Eigene Darstellung) Resultatanzeige

Durch Bestätigung des Spielers ruft die Resultatanzeige die Methode des mainscripts welche für das Zeichnen der Bananen zuständig ist, auf. Danach versteckt sie sich selbst und zeigt das Spielbrett wieder an.

23 Struktur

Die Aktivität befindet sich im Ordner Journeygame innerhalb des Sugar-build Verzeichnisses. Die Aktivität weist dabei folgende Ordnerstruktur auf.



Abbildung 73 : (Eigene Darstellung) Ordnerstruktur Journeygame

Folgende Elemente befinden sich in den Ordnern:

- activity

- In diesem Ordner befindet sich die activity.info Datei und die activity-icon.svg Datei.
- css
 - Hier befinden sich die CSS Datei die für die Vergabe von Styles verantwortlich ist.
- d3
 - Hier befindet sich die d3 Bibliothek
- images
 - Dieser Ordner beinhaltet alle Bilder die in der Aktivität verwendet wurden. Der Ordner ist in weitere Unterordner unterteilt. Die Unterteilung wurde gemäß Minispiele vorgenommen.
- index.html
 - Hier befindet sich sämtlicher HTML Code und somit die gesamte Struktur und Gliederung der Aktivität.
- JQuery
 - Hier befinden sich die JQuery Grundbibliotheken. In einem weiteren Unterordner sind hier auch die Bibliotheken der Plugins abgelegt.
- Js
 - In diesem Ordner befindet sich die gesamte Logik der Aktivität. Alle selbstgeschriebenen Javascripte wurden in diesem Ordner abgelegt.
- lib
 - In diesem Ordner befinden sich alle sugar-web Bibliotheken.
- LICENSE.txt
 - Dieses File beinhaltet alle Lizenzen
- Package.json
- Sounds
 - Hier befinden sich alle Sound Dateien. Die Sound Dateien sind alle im Ogg Vorbis Format.

24 Multimedia Dateien

24.1 Bilder

Für alle in der Arbeit verwendeten Bilder wurden vorgängig die Rechte eingeholt oder geprüft. Der Großteil der Bilder steht unter der Creative Common License. Für den kommerziellen Gebrauch verlangen einige einzelne Bilder einen Backlink. Diese Backlinks sind in der Aktivität über den Button in der oberen Leiste vorzufinden. Diese Backlinks lauten wie folgt:

- <http://www.visualpharm.com>
- <http://www.designcontest.com>
- <http://www.doublejdesign.co.uk>
- <http://artdesigner.lv>

Alle anderen Bilder wurden mit Hilfe von <http://search.creativecommons.org/> gesucht. Die Sucheinstellungen wurden so vorgenommen, dass nur Bilder welche für den kommerziellen Gebrauch verwendet werden dürfen angezeigt werden. Die Hintergrundbilder der Levels und sonstige Bilder wurden alle von mir selbst bearbeitet oder erstellt. Oft wurden zuvor gesuchte Bilder verwendet. Alle selbstangefertigte Bilder wurden mit Hilfe von GIMP und Photoscape X bearbeitet.

24.2 Sound

Alle verwendeten Sounddateien wurden mit Hilfe von <http://search.creativecommons.org/> gesucht oder aus der Online Sammlung von <http://soundbible.com/royalty-free-sounds-1.html> entnommen. Alle verwendeten Sounddateien sind unter der Creative Common Lizenz. Die Sounddateien wurden falls nötig online mit <http://cutmp3.net/de/> geschnitten. Da Sugar auf einem Linux Fedora basiert und dort die nötigen Bibliotheken zur MP3-Wiedergabe aus patentrechtlichen Gründen nicht installiert sind habe ich in meiner Bachelorarbeit Ogg Vorbis Dateien verwendet. Die Sounddateien wurden zuerst im MP3 Formt heruntergeladen und anschließend mit Hilfe der Webseite <http://audio.online-convert.com/convert-to-ogg> konvertiert.

25 Aventura Matemática auf dem OLPC Laptop

Wie in dieser Arbeit bereits erwähnt braucht es ein Update des OLPC Laptops um Sugar Aktivitäten erfolgreich auszuführen. Gemäß Christoph Guéret will OLPC in Zukunft auf Sugar Web setzen und arbeitet derzeit an einem Update. Meine Applikation kann sowohl in jedem Browser mit Hilfe von Sugarizer und auch in osbuild einer vollen Sugar Umgebung ausgeführt werden. In osbuild wird simuliert wie die Applikation nach dem Update eingesetzt werden kann. Im Rahmen meiner Bachelorarbeit möchte die Applikation aber auch auf einem OLPC Laptop präsentieren. Gemeinsam mit Herrn Johannes Krampf habe ich dazu folgende Lösung erarbeitet.

Zu Beginn meiner Arbeit stellte mir Fabian Cretton einen OLPC Laptop zur Verfügung. Dieser Laptop ist aufgrund des fehlenden Updates zwar nicht in der Lage die Sugar Aktivität direkt bei sich zu installieren und auszuführen. Allerdings verfügt der Laptop über einen Webbrowser und die Fähigkeit sich mit einem Wireless Local Area Network (W-LAN) Netzwerk zu verbinden. Folgende Abbildung zeigt die Erarbeitete Lösung um das Programm auf dem OLPC abzuspielen.

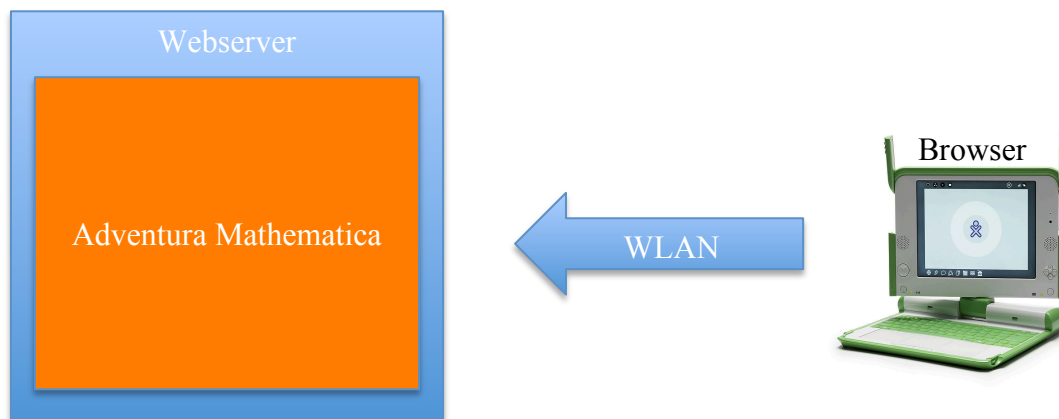


Abbildung 74 : (Eigene Darstellung) Aventura Mathematica über WLAN (Quelle OLPC Laptop: www.wikipedia.org, 2014)

Beim verwendeten Server handelt es sich um einen Windows Enterprise Server 2008. Der Server wurde von der Hes-so Wallis in Form einer virtuellen Maschine zur Verfügung gestellt und mit Hilfe des Programmes K-Trina über eine Virtual Private Network (VPN) Verbindung bezogen. Nach dem Download der virtuellen Maschine habe ich den Server als Webserver

konfiguriert. Dazu habe ich auf dem Server die entsprechenden Rollen installiert, die Ports konfiguriert und die nötigen Netzwerkeinstellungen vorgenommen. Damit der Server als eigenständiger Computer innerhalb des Netzwerks erkannt wird, muss der Netzwerkadapter dabei auf Bridged gestellt werden. Damit auch der Sound abgespielt werden kann muss ein benutzerdefinierter MIME Typ auf dem Server konfiguriert werden. Ansonsten erkennt der Internet Information Services (IIS) Server das ogg Format nicht. Mein konfigurierter Server ist nur in meinem eigenen Netzwerk zugänglich. Um die Applikation aber weltweit zugänglich zu machen hat Herr Krampf diese ebenfalls auf einem Server des Technopols abgelegt. Die Applikation kann durch folgenden Link erreicht werden. <http://phmf.hevs.ch/Journeygame/>.

Nach der Konfiguration des Webservers und dem erfolgreichen Verbindungsaufbau des OLPC Laptops mit dem Webserver musste ich feststellen, dass nicht alle Bilder und Sounddateien geladen werden konnten. Das Problem lag in der Größe aller Bilder und Sounddateien. Der Ordner der Bilder betrug 24,5 Megabyte (MB) und der Ordner mit den Sounddateien 37.0 MB. Um diesem Problem entgegenzuwirken reduzierte ich die Dateigröße der Bilder so dass der Bildordner schlussendlich lediglich noch 7.7 MB betrug. Um die Applikation erfolgreich auf dem OLPC Server zum Laufen zu bringen reduzierte ich zudem die Größe der Sounddateien sodass der Soundordner schlussendlich noch 9.7 MB betrug. Hierzu reduzierte ich die Kilobyte KB/s. Weiter wirke ich dem Problem durch aggressives Caching im Webbrowser entgegen. Für die Serverversion habe ich die Javascript Dateien komprimiert und in einer einzigen Datei zusammengefasst. Das Spiel funktioniert somit mit Ausnahme von einigen Sounds auf dem OLPC Laptop mit Hilfe der Browse Aktivität bereits einwandfrei.

Die Ausführung im Browser dient allerdings nur Demonstrationszwecken. Die Applikation soll nach dem Release des Updates für den OLPC als eigenständige Aktivität installiert und ausgeführt werden können. Wie untenstehende Abbildung zeigt lässt sich dies mit Hilfe von osbuild auf meiner virtuellen Maschine bereits simulieren.



Abbildung 75 : (Eigene Darstellung) Aventura Matematica unter osbuild

Somit wird sie performanter und stabiler als in der Browse Aktivität. Gemäß sugarlabs kann mit untenstehendem Befehl in osbuild ein XO bundle erstellt werden und auf dann auf den Sugar Activities Market hochgeladen werden.

```
python setup.py dist_xo
```

Quelle: (Developer Sugarlabs)

Meine Applikation wurde für den XO-4 implementiert. Gemeinsam mit Herrn Krampf habe ich auch versucht das Spiel auf einem XO-1 umzusetzen. Da die Browse Aktivität des XO-1 nicht alle Spielinhalte darstellen konnte, mussten wir zuerst das gesamte Betriebssystem sowie die Browse Aktivität updaten. Die Applikation läuft allerdings aufgrund der geringen Rechenpower des XO-1 sehr langsam. Der XO-1 verfügt nur über einen 433 Megahertz (MHz) Prozessor. Gemäß Absprache mit meinem Experten Laurent Sciboz hat die Optimierung des Programmes für einen XO-1 allerdings keine Priorität.

26 Git Version Control

Für die Versionskontrolle in meiner Arbeit habe ich Git verwendet. Git ist ein quelloffenes Versionskontrollsystem. Git eignet sich sowohl für kleine als auch größere Projekte. (git-scm.com, 2014)

Dazu wurde ein neues Repository mit dem Namen Aventura Mathematica angelegt. Untenstehende Grafik gibt über den Traffic, die Commit activity, die Code frequency und die Punch Card Auskunft.

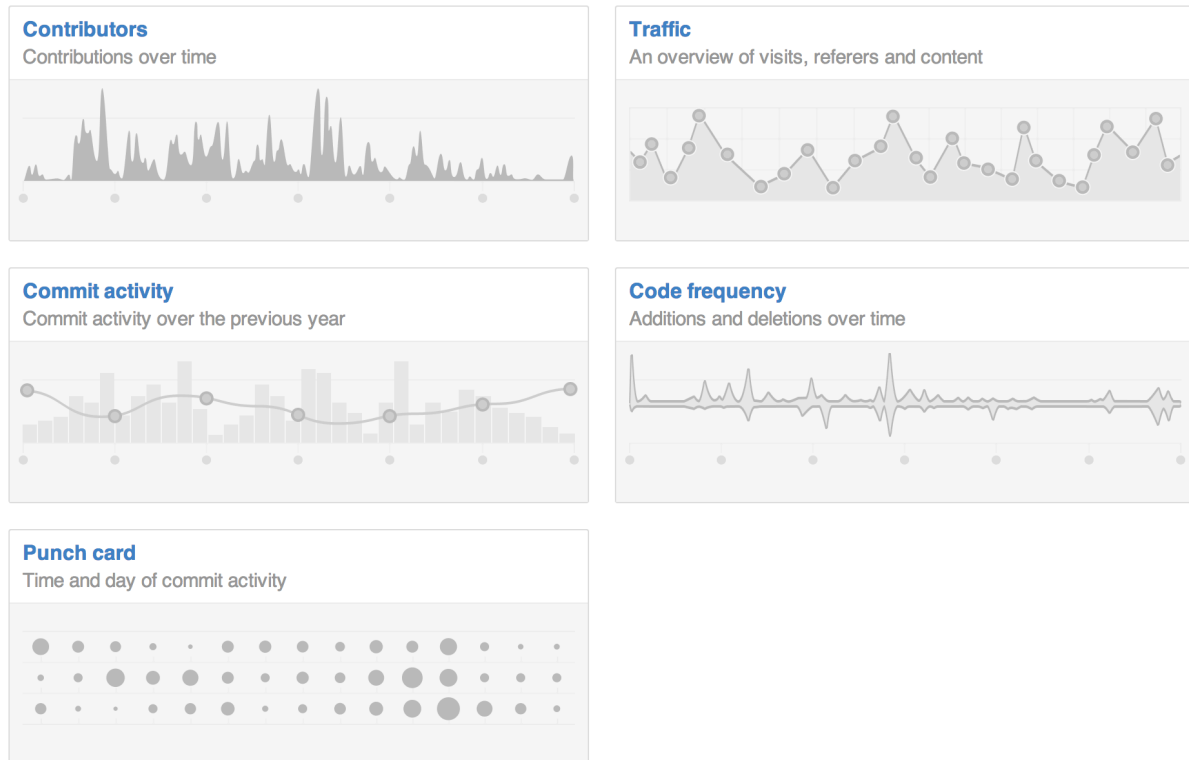


Abbildung 76 : Git Statistiken. Quelle: (Github.com, 2014)

27 Testing

Da für die Programmierung des Spiels Webtechnologien eingesetzt wurden entschied ich mich dazu die Tests manuell durchzuführen. Jede neu implementierte Funktion wurde von mir getestet. Bei den Tests habe ich folgendes Muster berücksichtigt:

- Testen der Grundfunktionalitäten
- Testen der Interaktion zwischen den verschiedenen Elementen
- Testen von Ausnahmefällen

Nach dem testen der Element wurde der Code mit Hilfe eines Commits auf Github geladen. Hinzu wurden bei verschiedenen Fertigungsstellungspunkten Programmversionen an Externe verteilt. Diese testeten die neuen Funktionen und gaben mir daraufhin hilfreiche Rückmeldung. Aufgrund der stetigen Arbeit am Spiel verlor ich den nötigen Abstand und war

für neue Anregungen sehr dankbar. Die Verbesserungsvorschläge der Testpersonen ließ ich in weitere Entwicklungen hineinfließen. Nach der Umsetzung der Verbesserungsvorschläge startete der Prozess des Testings wieder von vorne. Für weitere Tests stehen gemäß Angaben von Christoph Guéret eine Vielzahl von Leuten zur Verfügung.

28 Aktuelle Verfügbarkeit des Spiels

Die Aktivität ist bereits auf einem Server des Technopols in Siders abgelegt und unter dem Link <http://phmf.hevs.ch/Journeygame/> verfügbar. Die Applikation kann in jedem HTML 5 fähigen Browser ausgeführt werden. Aufgrund dessen, dass die Applikation für den OLPC Laptop konzipiert und implementiert wurde, müssen je nach Gerät noch zusätzliche Einstellungen getroffen werden um das Spiel reibungslos und ohne Verzerrungen der Oberfläche zum Laufen zu bringen. Künftig kann die Aktivität einfach auf den Sugar Market hochgeladen werden.

29 Abgabedokumente

Folgende Dokumente werden von mir zusätzlich zu diesem Bericht auf der CD abgegeben.

- ZIP Datei „Code Activity“
- ZIP Datei „Code Activity Compressed“
- ZIP Datei „Code Browser“
- ZIP Datei „Code Browser Compressed“
- Readme File

Auf dem USB-Stick befindet sich folgendes Image:

- Virtuelle Maschine Linux Fedora 19 Schrödingers Cat

Die ZIP Datei „Code Activity“ beinhaltet den Code des Spiels als eigenständige Aktivität. Das heißt der Aktivitätsbalken am oberen Rand ist in dieser Version integriert. Dieser Code wird gebraucht wenn das Spiel als eigenständige Aktivität, wie es in osbuild der Fall ist, ausgeführt wird. Dabei sind die JavaScript Dateien alle einzeln aufgeführt. Die ZIP Datei „Code Browser“ hingegen beinhaltet den Code der gebraucht wird wenn das Spiel zu Demonstrationszwecken mit Hilfe der Browse Aktivität und nicht als eigenständige Aktivität auf dem OLPC Laptop ausgeführt wird. Hier wird die Toolbar weggelassen da die Browse

Aktivität bereits über eine eigene Toolbar verfügt. Die beiden ZIP Dateien sind in Bezug auf die Logik identisch. Lediglich in Bezug auf die Styles gibt es Abweichungen da bei einer Version die Toolbar integriert ist und bei der anderen nicht.

Die ZIP Datei „Code Activity Compressed“ beinhaltet denselben Code wie die Datei „Code Activity“ in komprimierter Version. Ebenso beinhaltet die ZIP Datei „Code Browse Compressed“ den gleichen Code wie die Datei „Code Browse“ in komprimierter Version. In diesen komprimierten Versionen wurden alle 26 selbstgeschriebenen JavaScript Dateien in einer Datei zusammengefasst und eingebunden. Dazu wurden sämtliche Kommentare und Leerzeichen weggelassen. Auch die CSS Datei wurde komprimiert. Durch diese Komprimierung konnte die Performance gesteigert werden. Da der Code allerdings nicht mehr leserlich ist habe ich noch die lesbaren original Versionen „Code Activity“ und „Code Browse“ beigelegt. Die Komprimierung habe ich mit Hilfe des Online Tools YUI Compressor durchgeführt. Die ZIP Datei „Code Browse Compressed“ kann falls gewünscht entpackt und auf dem Server abgelegt werden und anschließend mit der Browse Aktivität des OLPC Laptops gestartet werden.

Die Virtuelle Maschine Linux Fedora 19 befindet sich im Gegensatz zu den anderen Dokumenten auf dem abgegebenen USB-Stick. Dies ist die Entwicklungsumgebung welche ich während meiner Arbeit verwendet habe. Hier kann die Aktivität nicht nur mit Hilfe von Sugarizer getestet werden, sondern auch innerhalb einer vollen Sugar Umgebung wie osbuild gestartet werden. Der Code der sich hier befindet ist in Bezug auf die Styles so angepasst, dass das Spiel als eigenständige Aktivität ausgeführt wird (die Toolbar integriert ist). Zudem wurde auch schon ein eigenes Aktivitätsicon erstellt und die Aktivität somit in Sugar einmalig gemacht. Unter osbuild auf der virtuellen Maschine kann die Aktivität mit Hilfe des Befehls „python setup.py dist_xo“ (Siehe Abschnitt Aventura Mathematica auf dem OLPC Laptop) falls gewünscht auf den Sugar Market hochgeladen werden.

Im Readme File befinden sich Erklärungen sowie sämtliche Passwörter für die virtuelle Maschine.

30 Schlussfolgerung

Die Analysephase meiner Bachelorarbeit zeigt die Probleme der Kinder in Dritt-Welt Ländern in Bezug auf die Bildung auf. Fundamentale Grundlagen wie Lesen und Rechnen werden von den Kindern nicht beherrscht. Dies hat verschiedene Faktoren wie beispielsweise die mangelnde Qualität des Unterrichts. Abgestimmte Lernprogramme können Helfen diesem Problem entgegenzuwirken. So können besonders die OLPC Laptops durch ihre Vielfalt an Lernprogrammen die Qualität des Unterrichts verbessern. Durch verschiedene Recherchen und den Austausch mit Experten des OLPC Projektes kristallisierte sich in meiner Arbeit heraus, dass OLPC in Zukunft vermehrt auf Web Technologien setzen wird. Durch Webtechnologien können die Spiele noch dynamischer und abwechslungsreicher implementiert werden.

Diese Bachelorarbeit war für mich eine einmalige Erfahrung. Für mich persönlich war es stets eine große Motivation ein Programm zu entwickeln, dass Kindern helfen kann. Dieser Gedanke wirkte stets beflügelnd und motivierend. Es war sehr spannend sich mit dem OLPC Projekt vertraut zu machen. Schon während der Analysephase konnte ich neue Erkenntnisse erlangen und sehr interessante Aufgaben ausführen. Die Konzeption und Implementierung des Spiels bereitete mir große Freude. Besonders die Konzeption des Spiels war anspruchsvoller als gedacht. In den einzusetzenden Technologien verfügte ich zu Beginn der Arbeit über Basiskenntnisse. Diese konnte ich im Verlaufe der Arbeit stark vertiefen. Teilweise hatte ich die Gelegenheit, Technologien welche ich nicht kannte einzusetzen. Das Spiel auf dem OLPC Laptop umzusetzen war ebenfalls interessant und herausfordernd.

Durch die Konzeption und Implementierung des Spiels „Aventura Matemática“ ist es mir gelungen ein Lernspiel für Kinder zu entwickeln. Meiner Meinung nach ist der Modulare Aufbau des Spiels das besonders schöne. Dadurch kann es stets weiterentwickelt werden. Jederzeit können neue Mini-Spiele zu dem bereits bestehenden Programm hinzugefügt werden. Diese können dabei unabhängig von der Applikation entwickelt und getestet werden. Danach können diese durch den wartbaren und übersichtlichen Code leicht in das bestehende Spiel eingebunden werden. So kann das Spiel für jede Stufe erweitert werden und auch verschiedene Themenbereiche abdecken. Das Spiel könnte durch passende Mini-Spiele nicht nur für den Mathematikbereich, sondern auch für andere Themen wie beispielsweise Geographie eingesetzt werden.

Das Spiel wurde als Sugar Web Aktivität implementiert. Aus diesem Grund kann es auf dem OLPC Laptop erst nach einem Update des Betriebssystems ausgeführt werden. In meiner Arbeit habe ich aber bereits die nötigen Vorkehrungen getroffen, so dass das Spiel schnell auf den Sugar Aktivitätsmarkt hochgeladen werden kann.

Abschließend möchte ich mich nochmal bei all den Personen die mich während der Arbeit unterstützt haben bedanken. Ich bin dankbar dass ich mich während meiner Bachelorarbeit mit einer so spannenden Thematik auseinandersetzen durfte. Ich hoffe dass die Kinder mit Hilfe meines Spiels viel lernen und ihrer Zukunft daraus einen Nutzen ziehen können.

31 Literaturverzeichnis

(26. 04 2014). (One Laptop per Child) Von wikipedia.org:
http://en.wikipedia.org/wiki/One_Laptop_per_Child abgerufen

(30. 04 2014). (UNESCO) Von unesco.de: http://www.unesco.de/bildung_fuer_alle.html
abgerufen

BMZ - Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung. (kein Datum). www.bmz.de. (B. -B. Entwicklung, Herausgeber) Abgerufen am 17. 04 2014 von www.bmz.de:
http://www.bmz.de/de/was_wir_machen/themen/bildung/hintergrund/bildungssituation/index.html?PHPSESSID=7a72b3156972e19f6966cc414d7bf8be

Developer Sugarlabs. (kein Datum). Abgerufen am 29. 04 2014 von Developer Sugarlabs:
<http://developer.sugarlabs.org/web-architecture.md.html>

Github.com. (01. 07 2014). Von <https://github.com/mbostock/d3/wiki> abgerufen
Github.com. (20. 07 2014). Von <https://github.com/kreuzerk/AdventuraMathematica/graphs>
abgerufen

git-scm.com. (04. 07 2014). Von <http://git-scm.com/> abgerufen

glade.gnome.org. (24. 06 2014). Von <https://glade.gnome.org/> abgerufen

Hinz, C. *Demografische Analysen - Konzepte - Strategien*. Berlin-Institut für Bevölkerung und Entwicklung.

Imsand, O. (30. 04 2014). Informationen zur Bildungslage in Peru. (K. Kreuzer, Interviewer, & K. Kreuzer, Herausgeber)

jquery.com. (02. 07 2014). Von <http://jquery.com/> abgerufen

laptop.org. (24. 06 2014). Von <http://wiki.laptop.org/go/Python> abgerufen

Massachusetts Curriculum Framework for Mathematics. (2011). *Grades Pre-Kindergarten to 12*. Massachusetts Curriculum Framework for Mathematics, Malden.

One Laptop per Child. (kein Datum). Abgerufen am 29. 04 2014 von One Laptop per Child:
<http://one.laptop.org/about/hardware>

One Laptop per Child. (kein Datum). Abgerufen am 20. 04 2014 von One Laptop per Child:
<http://one.laptop.org/about/xo-tablet>

One Laptop Per Child. (kein Datum). Abgerufen am 04. 29 2014 von One Laptop Per Child:
one.laptop.org

Popsci.com. (30. 04 2014). Von <http://www.popsci.com/gadgets/article/2013-07/one-laptop-childs-de-evolution> abgerufen

Sugarlabs. (kein Datum). *developer.sugarlabs.org*. Abgerufen am 6. 05 2014 von <http://developer.sugarlabs.org/activity.md.html>

Sugarlabs.org. (kein Datum). Abgerufen am 01. 05 2014 von <http://activities.sugarlabs.org/en-US/sugar/>

The World Wide Semantic Web. (kein Datum). Abgerufen am 04. 29 2014 von The World Wide Semantic Web: <http://worldwidesemanticweb.org/2014/04/16/impressions-from-sugarcamp3/>

UNESCO. (2014). *EFA Global Monitoring Report*. Education. France: UNESCO Publishing. Unesco. (01. 05 2014). *Unesco.de*. Von http://www.unesco.de/bildung_fuer_alle.html abgerufen

Unesco.org. (01. 05 2014). Von <http://www.unesco.org/new/en/education/themes/leading-the-international-agenda/efareport/infographics/out-of-school-children-globally/> abgerufen

Wiki Sugarlabs. (08. 05 2014). Von http://wiki.sugarlabs.org/go/Math4Team/Resources/Curriculum_Chart abgerufen

Wikipedia.org. (kein Datum). Abgerufen am 20. 04 2014 von Wikipedia.org: http://de.wikipedia.org/wii/OLPC_XO-1

www.spiegel.de. (30. 04 2014). (Der Spiegel) Von <http://www.spiegel.de/schulspiegel/ausland/unesco-bericht-weltweit-72-millionen-kinder-ohne-chance-auf-bildung-a-672821.html> abgerufen

www.wikipedia.de. (24. 06 2014). Von [http://de.wikipedia.org/wiki/Python_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Python_(Programmiersprache)) abgerufen

www.wikipedia.org. (04. 07 2014). Von http://en.wikipedia.org/wiki/OLPC_XO-1#mediaviewer/File:XO-Beta1-mikemcgregor-2.jpg abgerufen

www.zeit.de. (10. 10 1986). (Zeit Online) Abgerufen am 18. 04 2014 von [www.zeit.de](http://www.zeit.de/1986/42/mit-bildung-gegen-armut): <http://www.zeit.de/1986/42/mit-bildung-gegen-armut>

Anhang I: Interview mit Otto Imsand

Interview mit Otto Imsand
Naters, den 30. April 2014

Nachfolgendes Interview fand am 30. April 2014 17:00 Uhr in der Walliser Kanne in Naters statt. Das Interview erfolgte gemäß meiner Anfrage an Otto Imsand.

1. Guten Tag Herr Imsand, Könnten Sie mir kurz die wichtigsten Abschnitte Ihres Lebenslaufes auflisten?

Sehr gerne. Ich wurde 1963 in Naters geboren. Später besuchte ich das Lehrerseminar in Sion. Im Anschluss an das Seminar zog es mich an die Uni in Freiburg. Nach Abschluss des Studiums konnte ich einige Berufserfahrungen sammeln. Darunter war eine 2 jährige Anstellung in Andermatt. Danach begann für mich ein neues Lebenskapitel. Es zog mich nach Peru, genauer gesagt nach Lima. Hier lebte ich rund 17 Jahre. Ich arbeitete in Peru als Lehrer und unterrichtete am Colegio Pestalozzi.

2. Welche Fächer haben Sie unterrichtet?

Ich unterrichtete vorwiegend Deutsch, Mathematik, Englisch, Französisch und Naturlehre.

3. Was halten Sie vom OLPC Projekt?

Mir gefällt die Idee sehr gut. Ich finde das ganze Konzept sehr spannend und denke das es durchaus Potenzial hat.

4. Worauf sollte OLPC achten?

Meiner Meinung nach ist es wichtig die Eltern der Kinder vom Projekt zu überzeugen.

5. Welche Grundausbildung fehlt den Kindern in Peru oder Allgemein?

Besonders Lesen und Mathematik. Doch auch im Bereich des Allgemeinwissen herrscht großer Nachholbedarf. Z.b in Geographie oder Naturlehre mangelt es an Wissen. Lehrer sind auch nicht gut ausgebildet.

6. Woher kommt dieser Mangel an Wissen?

Das liegt daran, dass die Lehrkräfte in Peru nicht gut genug ausgebildet sind. Die Lehrpersonen verfügen nicht über das nötige Wissen und können dies den Kindern auch nicht richtig weitergeben.

7. Sie erwähnten vorhin, dass es den Kindern an Allgemeinwissen mangelt. Wozu brauchen die Kinder später das Allgemeinwissen. Gilt es nicht das Augenmerk auf Lesen und Schreiben zu legen?

Das Augenmerk sollte schon auf Lesen und Schreiben gelegt werden. Allerdings darf man nicht vergessen, dass die Grundausbildung auch die Basis für eine weiterführende Schule sein kann. Wer in Peru eine Universität absolviert, entkommt der Armut. Um allerdings in solche Universitäten zu gelangen muss man über ein gewisses Allgemeinwissen verfügen.

8. Wo liegen die Stärken der Kinder?

Die Kinder Perus sind auf jeden Fall sehr begeisterungsfähig. Darüber hinaus sind sie spontan. Im Gegensatz zu den Kindern und Jugendlichen hier sehen sie die Schule als eine Chance, als eine Art Ausweg aus der Armut und sind daher hochmotiviert.

9. Sie sagen die Kinder sind begeisterungsfähig, sollte man bei einem Lernprogramm gewisse Dinge beachten um diese „Begeisterungsfähigkeit“ zu wecken?

Auf jeden Fall. Bei einer Lernsoftware sollte darauf geachtet werden, dass die Inhalte farbig und abwechslungsreich gestaltet werden. Dabei sollte man möglichst viele Bilder verwenden. Lange Texte schrecken eher ab und bewegen die Kinder dazu die Applikation beiseite zu legen. Zudem ist es wichtig, dass die Kinder bei einer erfolgreich erledigten Aufgabe eine Belohnung und so gleichzeitig einen Anreiz erhalten.

10. Könnte man die Belohnung in den Spielen mit dem Allgemeinwissen kombinieren?

Ja. Auf jeden Fall. Das wäre eine sehr gute Idee.

11. Was gilt es in Hinblick auf die Sprache zu beachten?

Es gilt zu beachten, dass in Peru verschiedene Dialekte gesprochen werden. Grundsätzlich kann man normales Spanisch verwenden.

12. Wie wäre es wenn man eine Applikation hätte bei der der Lehrer selbst die Vokabeln erfassen könnte?

Damit würde man die Probleme mit der Sprache umgehen. Jeder Lehrer könnte die Inhalte auf die regionale Sprache bezogen gestalten. Weiter könnten auch die Lerninhalte auf regionale Objekte abgestimmt werden.

13. Wie kann ein Programm für die Kinder unterhaltsam sein und einen Lerneffekt hervorrufen?

Wie bereits gesagt ist es wichtig, dass die Kinder einen Anreiz haben das Spiel zu spielen. Daher ist es wichtig eine Art Belohnung oder Fortschritt einzubauen.

14. Wie sollten die Programminhalte gestaltet werden?

Die Programminhalte sollten so gestaltet werden, dass sie für die Kinder erkennbar und bekannt sind. Peru beispielsweise verfügt über drei grosse Regionen. Die Bergregion, die Küstenregion und den Urwald. Personen welche beispielsweise in den Bergen leben können mit dem Begriff Auto weniger anfangen als Personen welche an der Küste leben.

15. Wo liegen die Interessen der Kinder, wofür interessieren sich die Kinder in der Freizeit?
Die Kinder Perus, allgemein Südamerikas sind Fußball verrückt. Dies gilt nicht nur für die Jungen sondern auch für die Mädchen. In der Freizeit wird dort sehr viel Fußball gespielt.

16. Wie wichtig ist es die Hygiene ins Lehrmaterial einfließen zu lassen?
Hygiene ist ein sehr wichtiger Punkt. Besonders wichtig ist es die Kinder darauf aufmerksam zu machen das durch mangelnde Hygiene Krankheiten verursacht werden. In einigen Gebieten bringt es allerdings nicht viel wenn die Kinder wissen, dass sie Duschen sollten, dazu aber gar keine Möglichkeit haben.

Ich danke Ihnen vielmals für die interessanten Stellungnahmen.

Anhang II: Arbeitsjournal

Tag	Bezeichnung	Anzahl Stunden
Woche 1		
Montag 28.04.2014	<ul style="list-style-type: none"> Sitzung mit Laurent & Fabian. Einarbeitung in die Thematik Auflisten aller Länder + Anzahl Laptops Recherche Rapport + Planung Kontaktaufnahme mit Interviewpartnern 	09:00
Dienstag 29.04.2014	<ul style="list-style-type: none"> Recherche + Artikel Sugar Web State of the Art Recherche über „Needs“ Kontaktaufnahme DEZA Erstellung Layout Dokumentation Kontaktaufnahme Christoph Guéret 	08:00
Mittwoch 30.04.2014	<ul style="list-style-type: none"> Vorbereitung + Interview Vergleich OLPC und Tablet Durchführung Interview 	08:00
Donnerstag 01.05.2014	<ul style="list-style-type: none"> Überarbeitung von Texten Auswerten des Interviews 	01:00
Freitag 02.05.2014	<ul style="list-style-type: none"> Erster Entscheid OLPC Laptop vs OLPC Tablet Recherche OLPC Einsatzgebiete Auflisten der OLPC Einsatzgebieten Grafiken erstellen 	06:00
Total Woche 1		32:00
Woche 2		
Montag 05.05.2014	<ul style="list-style-type: none"> Recherche Bedürfnisse State of the Art auflisten Wichtigste Kapitel des Weltbildungsberichts herausfiltern 	08:00
Dienstag 06.05.2014	<ul style="list-style-type: none"> Weltbildungsbericht durcharbeiten 	07:00
Donnerstag 8.05.2014	<ul style="list-style-type: none"> Artikel über Bedürfnisse verfassen Softwareideen sammeln 	05:00
Freitag 09.05.2014	<ul style="list-style-type: none"> Statistiken berechnen Artikel über Bedürfnisse verfassen Vergleich der Technologien Softwareideen sammeln Spezifikation und Konzeption der neuen Applikation 	07:00
Total Woche 2		27:00
Woche 3		
Montag 12.05.2014	<ul style="list-style-type: none"> Sitzung Technopole. Präsentation Bedürfnis + Entscheid Applikation Vergleich der Technologien Entscheidung welche Technologie gebraucht wird 	08:00

Dienstag 13.05.2014	<ul style="list-style-type: none"> Analyse und Entscheid welche Entwicklungsumgebung gebraucht wird Selbststudium Linuxbefehle Aufsetzen der virtuellen Maschine – Linux Fedora 19 Schrödingers Cat Installation osbuild Vergleich IDE Aufsetzen der Entwicklungsugebung Recherche „Was wird im Bereich der Mathematik gebraucht?“ 	10:00
Mittwoch 14.05.2014	<ul style="list-style-type: none"> Installation git und Erstellung eines Git Repository Initial Commit Implementation Hello World Programm Git unter Fedora installieren Pfade richtig setzen 	02:00
Freitag 15.05.2014	<ul style="list-style-type: none"> Recherche welche Bibliotheken verwendet werden sollen Einarbeiten in svg und d3 Einarbeitung in JQuery 	08:00
Total Woche 3		28:00
Woche 4		
Montag 18.05.2014	<ul style="list-style-type: none"> Umgang mit GIMP auffrischen Erstellung Mock-up Level 1 Design Level 1 	08:00
Dienstag 19.05.2014	<ul style="list-style-type: none"> Zeichnen des Spielbretts mit Hilfe von d3 Bibliothek Buchstaben zentrieren JSON Daten richtig bereit stellen Überarbeitung Design Level 1 Charakter und Elemente mit svg zeichnen. Interaktionen der JavaScript Dateien modellieren Würfelwurf implementieren 	10:00
Freitag 22.05.2014	<ul style="list-style-type: none"> Charakterbewegung implementieren Ideensammlung Mini-Spiel 1 Entwurf Mini-Spiel 1 Beginn Implementierung Mini-Spiel 1 	08:00
Total Woche 4		26:00
Woche 5		
Montag 26.05.2014	<ul style="list-style-type: none"> Fertigstellung & Test Mini-Spiel 1 Integration Mini-Spiel 1 in die Applikation Konzeption Mini-Spiel 2 Beginn Mini-Spiel 2 	08:00
Dienstag 27.05.2014	<ul style="list-style-type: none"> Fertigstellung & Test Mini-Spiel 2 Beginn Mini-Spiel 3 Implementierung Mini-Spiel 3 	08:00
Freitag 30.05.2014	<ul style="list-style-type: none"> Fertigstellung & Test Mini-Spiel 3 Recherche Inputs neue Mini-Spiele Ideensammlung Mini-Spiele 	08:00

Total Woche 5		24:00
Woche 6		
Montag 02.06.2014	<ul style="list-style-type: none"> Erstellung Designs für Mini-Spiele Konzeption, Implementierung und Test Mini-Spiel 4. Testen der Applikation und Bugs beseitigen 	07:00
Dienstag 03.06.2014	<ul style="list-style-type: none"> Dokumentation Konzeption, Implementierung und Test Mini-Spiel 5. Konzeption und Design Mini-Spiel 6. Design und Entwurf Level 2 	09:00
Freitag 06.06.2014	<ul style="list-style-type: none"> Dokumentation Fertigstellung Mini-Spiel 6. Design der zum Level 2 zugehörigen Komponenten Kontaktaufnahme Mr. Guéret betreffend Update und Deployment Rechtliche Fragen betreffend Bilder abklären 	08:00
Total Woche 6		24:00
Woche 7		
Montag 09.06.2014	<ul style="list-style-type: none"> Dokumentation Auswählen aller Bilder Rechte auf neuen Bildern sicherstellen Erstellung und Design der Goldfelder 	07:00
Dienstag 10.06.2014	<ul style="list-style-type: none"> Ideensammlung neue Mini-Spiele Goldfelder in die Applikation integrieren Designs für Goldfelder erstellen Implementierung & Test Mini-Spiel 7 	08:00
Mittwoch 11.06.2014	<ul style="list-style-type: none"> Entwurf Mini-Spiel 8 	01:00
Freitag 14.06.2014	<ul style="list-style-type: none"> Integration Goldfelder in das Hauptspiel Implementierung & Test Mini-Spiel 8 Implementierung & Test Mini-Spiel 9 	10:00
Total Woche 7		26:00
Woche 8 (Modulprüfungswoche)		
Montag 17.06.2014	<ul style="list-style-type: none"> Dokumentation der Mini-Spiele Implementierung & Test Mini-Spiel 10 	04:00
Dienstag 18.06.2014	<ul style="list-style-type: none"> Verfassen des WB-Artikels 	01:00
Freitag 19.06.2014	<ul style="list-style-type: none"> Beginn Mini-Spiel 11 	02:00
Samstag 20.06.2014	<ul style="list-style-type: none"> Design Level 2 Implementierung & Test Mini-Spiel 11 Auswahlbildschirm Designs erstellen Logik für den Auswahlbildschirm erstellen 	08:00
Total Woche 8		15:00
Woche 9		
Montag 23.06.2014	<ul style="list-style-type: none"> Integration Level 2 Erstellung der JSON Positionsdaten für Level 2 Konzeption Mini-Spiel 13 	08:00
Dienstag 24.06.2014	<ul style="list-style-type: none"> Modellierung und Dok Mini-Spiele Implementierung & Test Mini-Spiel 13 	08:00
Mittwoch 25.06.2014	<ul style="list-style-type: none"> Implementierung & Test Mini-Spiel 14 	08:00

Donnerstag 26.06.2014	<ul style="list-style-type: none"> Erstellung von Sequenzdiagrammen Dokumentation Implementierung & Test Mini-Spiel 15 	05:00
Freitag 27.06.2014	<ul style="list-style-type: none"> Implementierung & Test Mini-Spiel 16 	06:00
Total Woche 9		35:00
Woche 10		
Montag 30.06.2014	<ul style="list-style-type: none"> Beschaffung Webserver Konfiguration Webserver (Porteinstellungen, Firewall-einstellungen und ändern von Policy) Verbindung mit OLPC Laptop 	06:00
Dienstag 01.07.2014	<ul style="list-style-type: none"> Optimierung des Programms für den OLPC Laptop Überarbeitung Dokumentation Programmtest auf dem OLPC durchgeführt 	06:00
Mittwoch 02.07.2014	<ul style="list-style-type: none"> Multimediaelemente komprimieren Performanceverbesserung Teilweise Integration von Lazy Loading 	08:00
Donnerstag 03.07.2014	<ul style="list-style-type: none"> Sounds für die Applikation bereitstellen Rechte auf Sounds sicherstellen 	08:00
Freitag 04.07.2014	<ul style="list-style-type: none"> Sounds komprimieren Beginn Implementierung Mini-Spiel 17 Design Level 3 	08:00
Total Woche 10		36:00
Woche 11		
Montag 07.07.2014	<ul style="list-style-type: none"> Fertigstellung & Test Mini-Spiel 17 Integration Level 3 JSON Daten Level 3 Dokumentation Goldfelder 	08:00
Dienstag 08.07.2014	<ul style="list-style-type: none"> Beginn Tiger Mini-Spiel Design Tiger Mini-Spiel 	08:00
Mittwoch 09.07.2014	<ul style="list-style-type: none"> Implementation & Test Logik Tiger Mini-Spiel Integration Tiger Mini-Spiel 	08:00
Donnerstag 10.07.2014	<ul style="list-style-type: none"> Sitzung Technopole mit Laurent Sciboz Integration von Mini-Game Resultaten 	08:00
Freitag 11.07.2014	<ul style="list-style-type: none"> Erstellung Lade Bildschirm Test + Bugs beseitigen 	08:00
Total Woche 11		40:00
Woche 12		
Montag 14.07.2014	<ul style="list-style-type: none"> Distribution Testprogramme an Testpersonen Nachdokumentieren von bereits implementierten Mini-Spielen Recherche neue Ideen Recherche neue JQuery Plugins 	08:00
Dienstag 15.07.2014	<ul style="list-style-type: none"> Endresultatanzeige einbinden Design anzeige gesperrte Levels Spielstände speichern Levels sperren Levels freischaltbar machen 	09:00

	<ul style="list-style-type: none"> • Memorygame Anzahl Versuche integrieren 	
Mittwoch 16.07.2014	<ul style="list-style-type: none"> • Wallgame Anzahl Versuche integrieren • Möglichkeit Spielstände zurückzusetzen • Rückmeldungen umsetzen • Verbesserungsvorschläge integrieren 	08:00
Donnerstag 17.07.2014	<ul style="list-style-type: none"> • Integration Link für die Quellenangabe • Link für Spielregeln integrieren • Integration „Espalda“ Button • Testen der Applikation • Beginn technische Dokumentation 	08:00
Freitag 18.07.2014	<ul style="list-style-type: none"> • Tests der Applikation auf dem OLPC Laptop • Technische Dokumentation • Testing 	08:00
Total Woche 12		41:00
Woche 13		
Montag 21.07.2014	<ul style="list-style-type: none"> • Code cleaning • Code kommentieren • Technische Dokumentation • Formatierungen gemäß Vorgabe • Testing + Bug fixing 	08:00
Dienstag 22.07.2014	<ul style="list-style-type: none"> • Formatierungen gemäß Vorgabe • Auswechseln der Bilder in der Dokumentation • Testing + Bugfixing 	08:00
Mittwoch 23.07.2014	<ul style="list-style-type: none"> • Korrekturen in der Dokumentation vornehmen • Online Dokument ausfüllen • Quellen überprüfen • An Urkund senden • Auf dem Laufwerk O ablegen. 	08:00
Donnerstag 24.07.2014	<ul style="list-style-type: none"> • Abgabedokumente bereitstellen – Drucken, Binden, Arbeit auf CD ablegen 	04:00
Freitag 25.07.2014	<ul style="list-style-type: none"> • Abgabe Bachelorarbeit 	
Total Woche 13		28:00
Gesamttotal		382:00

Anhang III – Quellcode mainscript.js

Dies ist nur ein Ausschnitt des Codes. Der gesamte Code ist unter folgender Adresse abrufbar.
<https://github.com/kreuzerk/AdventuraMathematica>

```
//Declaration of the variables
var currentfield = 1; //Field at the beginning
var movevalue;
var dicerresult;
var cx;
var cy;
var bananacounter = 0; //Bananamount at the beginning
var moneycounter = 0; //Moneyamount at the beginning
var filename;
var goldentered = false; //Checks if a golden field has been entered
var blackentered = false; //Checks if a black field has been entered
var endofround = false; //Checks if a round is over
var round = 1;
var dicethrown = false; //Checks if the dice has already been thrown
var deletearrow = true; //Checks if the arrow at the beginning needs to be
deleted
var rounds_played = 1; //Counts the number of played rounds
var entered; //Used to specify if the character has entered a field

//Variables that are needed at the end of the game

var number_correct_answered = 0;
var number_wrong_answered = 0;

//=====

var width = 1200;
var height = 820;

var circleData; //Stores all the positions for the selected level

//Circle Data Set for the first level
var level1 = [
  { "cx": 50, "cy": 770, "radius": 30, "color": "snow", "fontcolor": "black", "number": 1 },
  { "cx": 120, "cy": 700, "radius": 30, "color": "blue", "fontcolor": "white", "number": 2 },
  { "cx": 190, "cy": 700, "radius": 30, "color": "midnightblue", "fontcolor": "white",
    "number": 3 },
  { "cx": 260, "cy": 700, "radius": 30, "color": "silver", "fontcolor": "black", "number":
    4 },
  { "cx": 330, "cy": 720, "radius": 30, "color": "greenyellow", "fontcolor": "black",
    "number": 5 },
  { "cx": 400, "cy": 720, "radius": 30, "color": "green", "fontcolor": "white", "number":
    6 },
  { "cx": 470, "cy": 720, "radius": 30, "color": "red", "fontcolor": "white", "number": 7 },
  { "cx": 540, "cy": 720, "radius": 30, "color": "mediumaquamarine", "fontcolor": "black",
    "number": 8 },
  { "cx": 610, "cy": 720, "radius": 30, "color": "lavender", "fontcolor": "black", "number"
    : 9 },
  { "cx": 680, "cy": 720, "radius": 30, "color": "midnightblue", "fontcolor": "white",
    "number": 10 },
  { "cx": 750, "cy": 700, "radius": 30, "color": "lavender", "fontcolor": "black", "number"
    : 11 },
  { "cx": 820, "cy": 680, "radius": 30, "color": "red", "fontcolor": "white", "number":
    12 },
  { "cx": 890, "cy": 660, "radius": 30, "color": "silver", "fontcolor": "black", "number":
    13 },
  { "cx": 880, "cy": 590, "radius": 30, "color": "lavender", "fontcolor": "black", "number"
    : 14 },
```

```

    { "cx": 810, "cy": 580, "radius": 30, "color" : "greenyellow", "fontcolor" : "black",
"number" : 15},
    { "cx": 740, "cy": 580, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 16},
    { "cx": 670, "cy": 580, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
17},
    { "cx": 600, "cy": 580, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
18},
    { "cx": 530, "cy": 580, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 19},
    { "cx": 460, "cy": 580, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
20},
    { "cx": 390, "cy": 580, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 21},
    { "cx": 320, "cy": 580, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
22},
    { "cx": 320, "cy": 510, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
23},
    { "cx": 390, "cy": 500, "radius": 30, "color" : "turquoise", "fontcolor" : "black", "number"
: 24},
    { "cx": 460, "cy": 500, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
25},
    { "cx": 525, "cy": 500, "radius": 30, "color" : "salmon", "fontcolor" : "black", "number" :
26},
    { "cx": 590, "cy": 500, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
27},
    { "cx": 660, "cy": 500, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
28},
    { "cx": 730, "cy": 500, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
29},
    { "cx": 800, "cy": 500, "radius": 30, "color" : "chocolate", "fontcolor" : "white", "number"
: 30},
    { "cx": 870, "cy": 500, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
31},
    { "cx": 950, "cy": 480, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 32},
    { "cx": 1020, "cy": 460, "radius": 30, "color" : "lavender", "fontcolor" : "black", "number"
: 33},
    { "cx": 1070, "cy": 390, "radius": 30, "color" : "turquoise", "fontcolor" : "black",
"number" : 34},
    { "cx": 1090, "cy": 320, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
35},
    { "cx": 1070, "cy": 250, "radius": 30, "color" : "gold", "fontcolor" : "black", "number" :
36},
    { "cx": 1050, "cy": 180, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
37},
    { "cx": 980, "cy": 140, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
38},
    { "cx": 920, "cy": 110, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 39},
    { "cx": 850, "cy": 100, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
40},
    { "cx": 780, "cy": 100, "radius": 30, "color" : "chocolate", "fontcolor" : "white", "number"
: 41},
    { "cx": 550, "cy": 170, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
42},
    { "cx": 620, "cy": 170, "radius": 30, "color" : "salmon", "fontcolor" : "black", "number" :
43},
    { "cx": 690, "cy": 170, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
44},
    { "cx": 760, "cy": 170, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
45},
    { "cx": 830, "cy": 180, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 46},
    { "cx": 900, "cy": 190, "radius": 30, "color" : "turquoise", "fontcolor" : "black", "number"

```

```

: 47},
{ "cx": 970, "cy": 220, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
48},
{ "cx": 970, "cy": 290, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 49},
{ "cx": 920, "cy": 340, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
50},
{ "cx": 850, "cy": 340, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
51},
{ "cx": 780, "cy": 330, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
52},
{ "cx": 710, "cy": 330, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
53},
{ "cx": 640, "cy": 330, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
54},
{ "cx": 570, "cy": 330, "radius": 30, "color" : "black", "fontcolor" : "white", "number" :
55},
{ "cx": 500, "cy": 330, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
56},
{ "cx": 430, "cy": 330, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
57},
{ "cx": 360, "cy": 330, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 58},
{ "cx": 290, "cy": 350, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
59},
{ "cx": 220, "cy": 420, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
60},
{ "cx": 170, "cy": 490, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
61},
{ "cx": 130, "cy": 560, "radius": 30, "color" : "blue", "fontcolor" : "white", "number" :
62},
{ "cx": 140, "cy": 630, "radius": 30, "color" : "snow", "fontcolor" : "black", "number" :
63}];

//Datas for the second level
var level2 = [
{ "cx": 50, "cy": 770, "radius": 30, "color" : "snow", "fontcolor" : "black", "number" : 1
},
{ "cx": 70, "cy": 660, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 2},
{ "cx": 70, "cy": 590, "radius": 30, "color" : "khaki", "fontcolor" : "black", "number" :
3},
{ "cx": 70, "cy": 520, "radius": 30, "color" : "dodgerblue", "fontcolor" : "white", "number"
: 4},
{ "cx": 70, "cy": 450, "radius": 30, "color" : "salmon", "fontcolor" : "white", "number" :
5},
{ "cx": 70, "cy": 380, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
6},
{ "cx": 70, "cy": 310, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
7},
{ "cx": 70, "cy": 240, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 8},
{ "cx": 140, "cy": 210, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
9},
{ "cx": 210, "cy": 210, "radius": 30, "color" : "green", "fontcolor" : "white", "number"
:10},
{ "cx": 230, "cy": 280, "radius": 30, "color" : "chocolate", "fontcolor" : "white", "number"
: 11},
{ "cx": 240, "cy": 350, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
12},
{ "cx": 240, "cy": 420, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
13},
{ "cx": 240, "cy": 490, "radius": 30, "color" : "skyblue", "fontcolor" : "black", "number" :
14},
{ "cx": 240, "cy": 570, "radius": 30, "color" : "dodgerblue", "fontcolor" : "black",

```

```

"number" : 15},
{ "cx": 240, "cy": 640, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
16},
{ "cx": 310, "cy": 670, "radius": 30, "color" : "turquoise", "fontcolor" : "black", "number"
: 17},
{ "cx": 380, "cy": 670, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
18},
{ "cx": 450, "cy": 670, "radius": 30, "color" : "red", "fontcolor" : "black", "number" :
19},
{ "cx": 520, "cy": 670, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 20},
{ "cx": 590, "cy": 670, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
21},
{ "cx": 660, "cy": 670, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
22},
{ "cx": 730, "cy": 670, "radius": 30, "color" : "salmon", "fontcolor" : "black", "number" :
23},
{ "cx": 800, "cy": 670, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "white",
"number" : 24},
{ "cx": 870, "cy": 670, "radius": 30, "color" : "khaki", "fontcolor" : "black", "number" :
25},
{ "cx": 950, "cy": 670, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
26},
{ "cx": 950, "cy": 600, "radius": 30, "color" : "black", "fontcolor" : "white", "number" :
27},
{ "cx": 950, "cy": 530, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
28},
{ "cx": 880, "cy": 530, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
29},
{ "cx": 810, "cy": 500, "radius": 30, "color" : "chocolate", "fontcolor" : "white", "number"
: 30},
{ "cx": 860, "cy": 430, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
31},
{ "cx": 930, "cy": 400, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
32},
{ "cx": 1000, "cy": 350, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 33},
{ "cx": 1070, "cy": 300, "radius": 30, "color" : "gold", "fontcolor" : "black", "number" :
34},
{ "cx": 1070, "cy": 230, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
35},
{ "cx": 1000, "cy": 200, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
36},
{ "cx": 930, "cy": 180, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
37},
{ "cx": 860, "cy": 180, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
38},
{ "cx": 790, "cy": 160, "radius": 30, "color" : "skyblue", "fontcolor" : "black", "number" :
39},
{ "cx": 720, "cy": 180, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 40},
{ "cx": 650, "cy": 220, "radius": 30, "color" : "dodgerblue", "fontcolor" : "white",
"number" : 41},
{ "cx": 580, "cy": 240, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
42},
{ "cx": 510, "cy": 260, "radius": 30, "color" : "khaki", "fontcolor" : "black", "number" :
43},
{ "cx": 440, "cy": 280, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
44},
{ "cx": 370, "cy": 270, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
45},
{ "cx": 370, "cy": 200, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
46},
{ "cx": 440, "cy": 150, "radius": 30, "color" : "dodgerblue", "fontcolor" : "white",
"number" : 47},

```

```

    { "cx": 530, "cy": 150, "radius": 30, "color" : "salmon", "fontcolor" : "white", "number" :
48},
    { "cx": 600, "cy": 150, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
49},
    { "cx": 670, "cy": 120, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 50},
    { "cx": 720, "cy": 50, "radius": 30, "color" : "snow", "fontcolor" : "black", "number" :
51},];

//Datas for the third level
var level3 = [
    { "cx": 50, "cy": 770, "radius": 30, "color" : "snow", "fontcolor" : "black", "number" : 1
},
    { "cx": 120, "cy": 770, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
2},
    { "cx": 190, "cy": 770, "radius": 30, "color" : "purple", "fontcolor" : "white", "number" :
3},
    { "cx": 280, "cy": 770, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
4},
    { "cx": 350, "cy": 770, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
5},
    { "cx": 420, "cy": 770, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 6},
    { "cx": 490, "cy": 770, "radius": 30, "color" : "purple", "fontcolor" : "black", "number" :
7},
    { "cx": 520, "cy": 700, "radius": 30, "color" : "midnightblue", "fontcolor" : "white",
"number" : 8},
    { "cx": 530, "cy": 630, "radius": 30, "color" : "purple", "fontcolor" : "white", "number" :
9},
    { "cx": 520, "cy": 560, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :10},
    { "cx": 450, "cy": 520, "radius": 30, "color" : "pink", "fontcolor" : "black", "number" :
11},
    { "cx": 380, "cy": 520, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
12},
    { "cx": 310, "cy": 520, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
13},
    { "cx": 240, "cy": 520, "radius": 30, "color" : "gold", "fontcolor" : "black", "number" :
14},
    { "cx": 170, "cy": 520, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
15},
    { "cx": 120, "cy": 450, "radius": 30, "color" : "silver", "fontcolor" : "black", "number" :
16},
    { "cx": 120, "cy": 380, "radius": 30, "color" : "chocolate", "fontcolor" : "black", "number"
: 17},
    { "cx": 120, "cy": 310, "radius": 30, "color" : "khaki", "fontcolor" : "black", "number" :
18},
    { "cx": 120, "cy": 240, "radius": 30, "color" : "salmon", "fontcolor" : "black", "number" :
19},
    { "cx": 190, "cy": 240, "radius": 30, "color" : "purple", "fontcolor" : "white", "number" :
20},
    { "cx": 260, "cy": 240, "radius": 30, "color" : "orange", "fontcolor" : "black", "number" :
21},
    { "cx": 330, "cy": 240, "radius": 30, "color" : "yellow", "fontcolor" : "black", "number" :
22},
    { "cx": 400, "cy": 240, "radius": 30, "color" : "purple", "fontcolor" : "white", "number" :
23},
    { "cx": 470, "cy": 240, "radius": 30, "color" : "red", "fontcolor" : "white", "number" :
24},
    { "cx": 540, "cy": 210, "radius": 30, "color" : "turquoise", "fontcolor" : "black", "number"
: 25},
    { "cx": 610, "cy": 180, "radius": 30, "color" : "green", "fontcolor" : "white", "number" :
26},
    { "cx": 680, "cy": 150, "radius": 30, "color" : "mediumaquamarine", "fontcolor" : "black",
"number" : 27},
    { "cx": 700, "cy": 80, "radius": 30, "color" : "red", "fontcolor" : "white", "number" : 28},

```



```

    { "cx": 770, "cy": 60, "radius": 30, "color": "red", "fontcolor": "white", "number": 29},
    { "cx": 840, "cy": 60, "radius": 30, "color": "purple", "fontcolor": "black", "number": :
30},
    { "cx": 910, "cy": 60, "radius": 30, "color": "yellow", "fontcolor": "black", "number": :
31},
    { "cx": 1070, "cy": 150, "radius": 30, "color": "silver", "fontcolor": "black", "number": :
32},
    { "cx": 1000, "cy": 200, "radius": 30, "color": "mediumaquamarine", "fontcolor": "black",
"number": : 33},
    { "cx": 930, "cy": 190, "radius": 30, "color": "lavender", "fontcolor": "black", "number"
: 34},
    { "cx": 860, "cy": 190, "radius": 30, "color": "lavender", "fontcolor": "black", "number"
: 35},
    { "cx": 820, "cy": 260, "radius": 30, "color": "red", "fontcolor": "white", "number": :
36},
    { "cx": 820, "cy": 350, "radius": 30, "color": "khaki", "fontcolor": "black", "number": :
37},
    { "cx": 890, "cy": 350, "radius": 30, "color": "orange", "fontcolor": "black", "number": :
38},
    { "cx": 960, "cy": 350, "radius": 30, "color": "black", "fontcolor": "white", "number": :
39},
    { "cx": 1030, "cy": 350, "radius": 30, "color": "salmon", "fontcolor": "black", "number": :
40},
    { "cx": 1080, "cy": 400, "radius": 30, "color": "midnightblue", "fontcolor": "white",
"number": : 41},
    { "cx": 1080, "cy": 470, "radius": 30, "color": "greenyellow", "fontcolor": "black",
"number": : 42},
    { "cx": 1080, "cy": 540, "radius": 30, "color": "orange", "fontcolor": "black", "number": :
43},
    { "cx": 1080, "cy": 610, "radius": 30, "color": "green", "fontcolor": "white", "number": :
44},
    { "cx": 1010, "cy": 650, "radius": 30, "color": "khaki", "fontcolor": "black", "number": :
45},
    { "cx": 940, "cy": 640, "radius": 30, "color": "greenyellow", "fontcolor": "black",
"number": : 46},
    { "cx": 870, "cy": 630, "radius": 30, "color": "midnightblue", "fontcolor": "white",
"number": : 47},
    { "cx": 800, "cy": 630, "radius": 30, "color": "orange", "fontcolor": "black", "number": :
48},
    { "cx": 780, "cy": 560, "radius": 30, "color": "green", "fontcolor": "white", "number": :
49},
    { "cx": 710, "cy": 540, "radius": 30, "color": "midnightblue", "fontcolor": "white",
"number": : 50},
    { "cx": 640, "cy": 540, "radius": 30, "color": "red", "fontcolor": "white", "number": :
51},
    { "cx": 640, "cy": 610, "radius": 30, "color": "green", "fontcolor": "white", "number": :
52},
    { "cx": 640, "cy": 680, "radius": 30, "color": "yellow", "fontcolor": "black", "number": :
53},
    { "cx": 640, "cy": 750, "radius": 30, "color": "snow", "fontcolor": "black", "number": :
54}];

/*=====
Draw the gameboard
=====*/
function drawgameboard(){

//Add circles to the svgContainer
var circles = svgContainer.selectAll("circle")
    .data(circleData)
    .enter()
    .append("circle");

//Add the circle attributes
var circleAttributes = circles

```

```

        .attr("cx", function (d) { return d.cx; })
        .attr("cy", function (d) { return d.cy; })
        .attr("r", function (d) { return d.radius; })
        .style("fill", function (d) { return d.color;
    }).attr("stroke","black");

//Add the SVG Text Element to the svgContainer
var text = svgContainer.selectAll("text")
    .data(circleData)
    .enter()
    .append("text");

//Add SVG Text Element Attributes
var textLabels = text
    .attr("x", function(d) { if(d.number < 10){return d.cx-5;}else{ return
d.cx-10;} })
    .attr("y", function(d) { return d.cy+5; })
    .text(function(d) { return d.number; })
    .attr("font-family", "Chalkboard")
    .attr("font-size", "20px")
    .attr("fill", function(d) { return d.fontcolor; });

//Adding an image
svgContainer.append("svg:image")
    .attr("xlink:href", filename).attr("id", "Lion")    //Filename is specified in the
mainscript
    .attr("width", 100)
    .attr("height", 100).attr("x", 0).attr("y",680);

//Adding the image in the top right cornert - imgae to go back to the main game
svgContainer.append("svg:image")
    .attr("onclick","espalda_clicked()")
    .attr("xlink:href", "./images/paper_corner.png")
    .attr("width", 260)
    .attr("height", 120).attr("x", 1000).attr("y",0);

//Adding the cube
svgContainer.append("svg:image")
    .attr("xlink:href", "./images/cube.png").attr("onclick","throwdice()").attr("id",
"maingame_cube")
    .attr("width", 100)
    .attr("height", 100).attr("x", 20).attr("y",20);

//Adding the arrow that shows that you need to click on the dice
svgContainer.append("svg:image").attr("id", "click_arrow")
    .attr("xlink:href", "./images/arrow.png")
    .attr("width", 200)
    .attr("height", 200).attr("x", 150).attr("y",0);

//=====
//Section at the RIGHT BOTTOM
//=====

//Adding the bananaimage
svgContainer.append("svg:image")
    .attr("xlink:href", "./images/bananas.png")
    .attr("width", 80)
    .attr("height", 80).attr("x", width-130).attr("y",height-120);

//Adding the Bananatext
svgContainer.append("text").attr("id", "bananaamount")
    .attr("x", width-240)
    .attr("y", height-50)

```

```

        .text("0 :")
        .attr("font-family", "Chalkboard")
        .attr("font-size", "50px")
        .attr("fill", item_display_color);

//Adding the moneyimage
svgContainer.append("svg:image")
    .attr("xlink:href", "../images/money.png")
    .attr("width", 80)
    .attr("height", 80).attr("x", width-360).attr("y", height-120);

//Adding the moneytext
svgContainer.append("text").attr("id", "moneyamount")
    .attr("x", width-440)
    .attr("y", height-50)
    .text("0 :")
    .attr("font-family", "Chalkboard")
    .attr("font-size", "50px")
    .attr("fill", item_display_color);

}

/*=====
Function that is used to set the filename of the character - called by the selectionscreen.js
=====*/
/
function setfilename($file){
    filename = $file;           //Set the filepath of the character
}

/*=====
Function that is called when continuebutton on the startscreen is clicked
=====*/
function selectionscreen_continueclicked(){

$("#startscreen").hide();
sound_introeffect.play();
sound_intromusic.play();
$("#characterselection").show();

}

/*=====
Throw the dice
=====*/
function throwdice(){

sound_diceroll.play(); //Play the sound for the diceroll

if(deletearrow){           //Defines if the arrow at the begining needs to be deleted
    var arrow = document.getElementById("click_arrow");
    arrow.parentNode.removeChild(arrow);
    deletearrow = false;
}

if(dicethrown == false){   //Throw the dice only when the character is not moving

if(entered){ //Call Methods at the beginning of each dice throw
    remove();
}

diceresult = Math.floor(Math.random() * 6) + 1; //Simulates the dice throw

var filename;
//Draw the corresponding image to the diceresult

```

```

switch(diceresult){
case 1:
    filename = "One";
break;
case 2:
    filename = "Two";
break;
case 3:
    filename = "Three";
break;
case 4:
    filename = "Four";
break;
case 5:
    filename = "Five";
break;
case 6:
    filename = "Six";
break;
}

var manuscript_operators = ['+', '-', '*', '/'];

//Change the level_operator
switch(level_operator){

case "+":
    level_operator = "-";
break;
case "-":
    level_operator = "+";
break;
case "*":
    level_operator = "/";
break;
case "/":
    level_operator = "*";
break;

}

drawcube(filename);

movevalue=currentfield;
move();
} //End if
}

/*=====
Function that is responsible for moving the character
=====*/
function move(){

    dicethrown = true;

    if(movevalue<(currentfield+diceresult) && goldentered == false && endofround == false &&
blackentered == false){
        window.setTimeout("move()",1000);
        movevalue++;

        cx = circleData[movevalue-1].cx; //Read the x coordinate of the next field
        cy = circleData[movevalue-1].cy; //Read the y coordinate of the next field

        //Delete the lion
        delete_character();
    }
}

```

```

//Draw the lion
draw_character(cx,cy);

if(circleData[movevalue-1].color == "gold"){ //Detect if we entered a gold field
    goldentered = true;
}

if(circleData[movevalue-1].color == "black"){ //Detect if we entered a black field
    blackentered = true;
}

if(circleData[movevalue-1].number >= circleData.length){ //Detect if we are at the
End of the round
    endofround = true;
}
}
else{

//Needs to be the first ifStatement otherwise it will not work
if(!goldentered && !endofround && !blackentered){
    currentfield = dicerresult + currentfield;
}

if(blackentered){ //The character has passed a black field
    currentfield = movevalue;
    blackentered = false; //Because we are going to leave the gold field
}

if(endofround == true){ //The round is over
    currentfield = 1; //Set the next field to 1
    endofround = false;
}

if(goldentered){
    currentfield = movevalue;
    goldentered = false; //Because we are going to leave the gold field
}

removedicedisplay();//Remove the dicatedisplay
checkfield(); //Check the field we are standing on
dicethrown = false;
}
}

/*=====
Function that is responsible for drawing the lion
=====*/
function draw_character($cx, $cy){

//Play the move sound
sound_blop.play();

//Append the lion to the svg container
svgContainer.append("svg:image")
    .attr("xlink:href", filename).attr("id", "Lion")
    .attr("width", 100)
    .attr("height", 100).attr("x", $cx-50).attr("y",$cy-90);
}

/*=====
Function that is responsible for deleteing the lion
=====*/
function delete_character(){

```

```

var lion = document.getElementById("Lion");
lion.parentNode.removeChild(lion);
}

/*=====
Function that is responsible for drawing the cube
=====*/
function drawcube($filename){

//Draw the texts and the cube according to the currentfield and the thrown number

svgContainer.append("svg:image")
    .attr("xlink:href", "./images/" + $filename + ".png").attr("id", "diceimage")
    .attr("width", 100)
    .attr("height", 100).attr("x", 290).attr("y",30);

svgContainer.append("text").attr("id", "dicetextone")
    .attr("x", 160)
    .attr("y", 100)
    .text(currentfield + " + ")
    .attr("font-family", "Chalkboard")
    .attr("font-size", "50px")
    .attr("fill", item_display_color);

svgContainer.append("text").attr("id", "diceresult")
    .attr("x", 420)
    .attr("y", 100)
    .text("=" + (currentfield+diceresult))
    .attr("font-family", "Chalkboard")
    .attr("font-size", "50px")
    .attr("fill", item_display_color);

}

/*=====
Remove the dice display in the middle of the window
=====*/
function removedicedisplay(){

var dicetextone = document.getElementById("dicetextone");
var diceimage = document.getElementById("diceimage");
var diceresult = document.getElementById("diceresult");

//remove the elements
dicetextone.parentNode.removeChild(dicetextone);
diceimage.parentNode.removeChild(diceimage);
diceresult.parentNode.removeChild(diceresult);
}

/*=====
Hide function at the beginning of the game - Hide the not used elements
=====*/
function hide(){

$("#shopdiv").hide();
$("#maingamesvg").hide();
$("#krokodiv").hide();
$("#formgame").hide();
$("#endresultgame").hide();
$("#whatisnextgame").hide();
$("#cheese").hide();
$("#tengame").hide();
$("#divisiongame").hide();
$("#bananadelivery").hide();

```

```

$("#slidergame").hide();
$("#wallgame").hide();
$("#endofround").hide();
$("#sourcediv").hide();
$("#circlegame").hide();
$("#multiplikationtablediv").hide();
$("#gridaddition").hide();
$("#sharinggame").hide();
$("#memorygame").hide();
$("#castlegame").hide();
$("#bossquiz").hide();
$("#tigerdiv").hide();
$("#progress_buttondiv").hide();
$("#characterselection").hide();
$("#newleveldiv").hide();
$("#gamerules").hide();
$("#minigameresultdiv").hide();
$("#adventuregame_endresultdiv").hide();
}

/*=====
Function that detects on which field the character is standing
=====*/
function checkfield(){

var currentcolor = circleData[currentfield-1].color;    //Read the current color the
character is standing on

//Stop the sound corresponding to the level
switch(selected_level){

case "level1":
sound_backgroundmusic_level1.pause();
break;
case "level2":
sound_level2.pause();
break;
case "level3":
sound_level3.pause();
break;

}

//Say which method needs to be called. Corresponding to the color the character is standing on.
switch(currentcolor){
case "green":
green();
break;
case "red":
red();
break;
case "blue":           //Blue = Shop Game
//Hide the main game
$("#maingamesvg").hide();
sound_background2.play(); //Play the background music for the shopgame
$("#shopdiv").show();
createexercise();
break;
case "yellow":         //Yellow = Krokogame
$("#maingamesvg").hide();
createkrokogame();
sound_background_krokogame.play();
$("#krokodiv").show();
break;
case "orange":         //Orange == formgame

```

```

$("#maingamesvg").hide();
sound_background2.play();
createformsgame();
$("#formgame").show();
break;
case "purple": //Purple == Endresultgame
$("#maingamesvg").hide();
sound_background2.play();
createendresultgame();
$("#endresultgame").show();
break;
case "pink": //Pink == whatisnextgame
$("#maingamesvg").hide();
sound_background2.play();
nextsvg_drawexercise();
$("#whatisnextgame").show();
break;
case "lavender": //Lavender == Cheesegame
$("#maingamesvg").hide();
sound_background2.play();
initialisecheesegame();
$("#cheese").show();
break;
case "skyblue": //Skyblue == Tengame
$("#maingamesvg").hide();
sound_background2.play();
initialisetennumber();
$("#tengame").show();
break;
case "mediumaquamarine": //Mediumaquamarine == Sortgame
$("#maingamesvg").hide();
sound_background2.play();
initialisedivisiongame();
$("#divisiongame").show();
break;
case "gold": //Gold - Dliwer the bananas
$("#maingamesvg").hide();
sound_monkey.play();
sound_monkeybackground.play();
initialise_delivery_screen();
$("#bananadelivery").show();
break;
case "khaki": //Khaki - Slidergame
$("#maingamesvg").hide();
sound_background2.play();
initialiseslidergame();
$("#slidergame").show();
break;
case "midnightblue": //Midnightblue fields - Wallgame
$("#maingamesvg").hide();
sound_background2.play();
initialisewallgame();
$("#wallgame").show();
break;
case "snow":
if(rounds_played == adventuregame_number_of_rounds){ //The game is finished
$("#maingamesvg").hide();
initialise_endresultscreen();
$("#adventuregame_endresultdiv").show();
}
else{
rounds_played++;
$("#maingamesvg").hide();
$("#endofround").show();
}
}

```



```

break;
case "turquoise": //Turquoise fields - Circlegame
    $("#maingamesvg").hide();
    sound_background2.play();
    initialisecirclegame();
    $("#circlegame").show();
break;
case "salmon": //Salmon fields - Mathtable
    $("#maingamesvg").hide();
    sound_background2.play();
    initialise_math_table();
    $("#multiplikationtablediv").show();
break;
case "greenyellow": //Greenyellow fields => Grid addition
    $("#maingamesvg").hide();
    sound_background2.play();
    initialise_grid_add_game();
    $("#gridaddition").show();
break;
case "dodgerblue": //Dodgerblue fields => Share game
    $("#maingamesvg").hide();
    initialise_sharinggame();
    sound_background2.play();
    $("#sharinggame").show();
break;
case "chocolate": //Chocolate fields => Memorygame
    $("#maingamesvg").hide();
    sound_background2.play();
    initialise_memory_game();
    $("#memorygame").show();
break;
case "silver": //Silver fields => Castlegame
    $("#maingamesvg").hide();
    sound_castlebackground.play();
    castlegame_resetlives();
    castlegame_initialise();
    $("#castlegame").show();
break;
case "black": //Check the black fields
    $("#maingamesvg").hide();
    sound_tigerbackground1.play();
    initialise_tiger_div();
    $("#tigerdiv").show();
break;
}
}

/*=====
Function for green fields
=====*/
function green(){
    minigameresult_greenfields(3, true);

    switch(selected_level){
    case "level1":
        sound_backgroundmusic_level1.pause();
    break;
    case "level2":
        sound_level2.pause();
    break;
    case "level3":
        sound_level3.pause();
    break;
    }
}

```

```

}

$("#maingamesvg").hide();
$("#minigameresultdiv").show();

}

/*=====
Remove the images from the display
=====*/
function remove(){

var plusbananaimage = document.getElementById("bananaimage");
var plusbananatext = document.getElementById("bananatext");

//remove the elements
plusbananaimage.parentNode.removeChild(plusbananaimage);
plusbananatext.parentNode.removeChild(plusbananatext);

entered = false;
}

/*=====
Function for red fields
=====*/

//Display function
function red(){
minigameresult_redfields(3, false);

switch(selected_level){
case "level1":
sound_backgroundmusic_level1.pause();
break;
case "level2":
sound_level2.pause();
break;
case "level3":
sound_level3.pause();
break;
}

$("#maingamesvg").hide();
$("#minigameresultdiv").show();
}

/*=====
Function for blue fields (Shopgame)
=====*/

function correctblue(){
initialiseminigameresultscreen(10, true);
hideblue();
}

function wrongblue(){
initialiseminigameresultscreen(10, false);
hideblue();
}

function hideblue(){
$("#shopdiv").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

```

```

}

/*=====
Function for yellow fields (Krokogame)
=====*/

function checkkrokoanswers($number_correct){

switch($number_correct){

case 0:
    initialiseminigameresultscreen(10, false);
break;
case 1:
    initialiseminigameresultscreen(5, true);
break;
case 2:
    initialiseminigameresultscreen(10, true);
break;
case 3:
    initialiseminigameresultscreen(15, true);
break;
}

hideyellow();

}

function hideyellow(){
    $("#krokodiv").hide();
    sound_background_krokogame.pause();
    $("#minigameresultdiv").show();
}

/*=====
Function for orange fields
=====*/

function checkorangefields($correct){

if($correct){
    initialiseminigameresultscreen(10, true);
}
else{
    initialiseminigameresultscreen(10, false);
}

hideorange();

}

function hideorange(){
    $("#formgame").hide();
    sound_background2.pause();
    $("#minigameresultdiv").show();
}

/*=====
Function for purple fields
=====*/

function checkpurplefields($correct){
if($correct){
    initialiseminigameresultscreen(20, true);
}
}

```

```

else{
    initialiseminigameresultscreen(5, false);
}

hidepurple();

}

function hidepurple(){
$("#endresultgame").hide();
    sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for pink fields
=====*/
function checkpinkfields($is_correct){

hidepink();

if($is_correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

}

function hidepink(){
$("#whatnextgame").hide();
    sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for lavender fields
=====*/
function checklavenderfields($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

}

hidelavender();

}

function hidelavender(){
$("#cheese").hide();
    sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for skyblue fields
=====*/
function checkskybluefields($is_correct){

```

```

hideskyblue();

if($is_correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

}

function hideskyblue(){
$("#tengame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for mediumaquamarine fields
=====*/
function checkmediumaquamarinefields($is_correct){

hidemediumaquamarine();

if($is_correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

}

function hidemediumaquamarine(){
$("#divisiongame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for golden fields
=====*/
function checkgoldenfields($deliver){

if($deliver){
    initialiseminigame_money_recived(true);
}
else
{
    initialiseminigame_money_recived(false);
}

hidegold();

}

function hidegold(){
$("#bananadelivery").hide();
sound_monkeybackground.pause();
$("#minigameresultdiv").show();
}

/*=====

```

```

Function for khaki fields
=====*/
function checkkhakifields($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

hidekhaki();

}

function hidekhaki(){
$("#slidergame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for midnightblue fields
=====*/
function checkmidnightbluefields($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
else
{
    initialiseminigameresultscreen(10,false);
}

hidekmidnightblue();

}

function hidekmidnightblue(){
$("#wallgame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for snow fields
=====*/
function hidesnowfield(){
$("#endofround").hide();
initialiseminigameresultscreen(10,true);
sound_buttonclicked.play();
$("#minigameresultdiv").show();
}

/*=====
Function for turqois fields
=====*/
function checkturquoisfields($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
else
{
    initialiseminigameresultscreen(10, false);
}

```

```

}

hideturquois();

}

function hideturquois(){
$("#circlegame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function to check the salmon fields
=====*/
function checksalmonfields($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
else
{
    initialiseminigameresultscreen(10, false);
}

hidesalmon();

}

function hidesalmon(){
$("#multiplikationtablediv").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function to check the greenyellow fields
=====*/
function checkgreenyellowfields($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

hidegreenyellow();

}

function hidegreenyellow(){
$("#gridaddition").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function to check the dodgerblue fields
=====*/
function checkdodgerblue($correct){

if($correct){
    initialiseminigameresultscreen(10,true);
}
}

```

```

else
{
    initialiseminigameresultscreen(10, false);
}

hidedodgerblue();

}

function hidedodgerblue(){
$("#sharinggame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function to check the chocolate fields
=====*/
function checkchocolate($correct, $number){

if($correct){
    initialiseminigameresultscreen($number,true);
}
else
{
    initialiseminigameresultscreen($number, false);
}

hidechocolate();

}

function hidechocolate(){
$("#memorygame").hide();
sound_background2.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for silver fields
=====*/
function checksilver($is_correct){

if($is_correct){
    initialiseminigameresultscreen(10,true);
}
else{
    initialiseminigameresultscreen(10,false);
}

hidesilver();

}

function hidesilver(){
$("#castlegame").hide();
sound_castlebackground.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for tiger game      / Black fields
=====*/
function checkblackfields($result){

```



```

switch($result){

case -1:
initialiseminigameresultscreen(30, false);
break;
case 0:
initialiseminigameresultscreen(10, false);
break;
case 1:
initialiseminigameresultscreen(0, true);
break;

}

hideblack();

}

function hideblack(){
$("#tigerdiv").hide();
sound_tigerbackground1.pause();
$("#minigameresultdiv").show();
}

/*=====
Function for drawing the banana
=====*/
function drawimage($amount, $positive){

//Play the sound corresponding to the level
switch(selected_level){

case "level1":
sound_backgroundmusic_level1.play();
break;
case "level2":
sound_level2.play();
break;
case "level3":
sound_level3.play();
break;

}

//Display banana
svgContainer.append("svg:image")
    .attr("xlink:href", "./images/bananas.png").attr("id", "bananaimage")
    .attr("width", 100)
    .attr("height", 100).attr("x", 290).attr("y",30);

if($positive){

//Plus
svgContainer.append("text").attr("id", "bananatext")
    .attr("x", 160)
    .attr("y", 100)
    .text(" + " + $amount)
    .attr("font-family", "Chalkboard")
    .attr("font-size", "50px")
    .attr("fill", item_display_color);

var bananaamount = document.getElementById("bananaamount");
bananaamount = bananaamount + $amount;

```

```

bananaamount.textContent = bananacounter + " :";
}
else{

//Minus
svgContainer.append("text").attr("id", "bananatext")
    .attr("x", 160)
    .attr("y", 100)
    .text("- " + $amount)
    .attr("font-family", "Chalkboard")
    .attr("font-size", "50px")
    .attr("fill", "red");

var bananaamount = document.getElementById("bananaamount");
if(bananacounter - $amount < 0){
    bananacounter = 0;
}
else{
    bananacounter = bananacounter - $amount;
}
}

bananaamount.textContent = bananacounter + " :";
entered = true;
}

/*=====
Function for drawing the money
=====*/
function drawmoney($amount, $positive){

//Display banana
svgContainer.append("svg:image")
    .attr("xlink:href", "../images/money.png").attr("id", "bananaimage")
    .attr("width", 100)
    .attr("height", 100).attr("x", 290).attr("y", 30);

if($positive){

//Plus
svgContainer.append("text").attr("id", "bananatext")
    .attr("x", 160)
    .attr("y", 100)
    .text("+ " + 1)
    .attr("font-family", "Chalkboard")
    .attr("font-size", "50px")
    .attr("fill", item_display_color);

var moneyamount = document.getElementById("moneyamount");
moneycounter = moneycounter + $amount;
moneyamount.textContent = "+ " + moneycounter;

if(moneycounter >= 3){

switch(selected_level){

case "level1":

if(level != "2" && level != "3"){
    localStorage.setItem("level", 2); //Set the amount of levels
    document.getElementById("unlockedtext").innerHTML = "Has desbloqueado Universo Matemático";
    sound_newlevel.play();

```

```

level = "2";
$("#newleveldiv").dialog("open");
}
break;
case "level2":
if(level != "3"){
localStorage.setItem("level", 3);
level = "3";
document.getElementById("unlockedtext").innerHTML = "Has desbloqueado Viejo Oeste Matemático";
sound_newlevel.play();
$("#newleveldiv").dialog("open");
}
break;
case "level3":
if(charakter_locked == "0"){ //Check if the character is already playable
localStorage.setItem("character", 1); //Unlock the character
charakter_locked = 1;
document.getElementById("unlockedtext").innerHTML = "Has desbloqueado hipopótamo";
sound_newlevel.play();
$("#newleveldiv").dialog("open");
}
break;
}
}

//Reduce the bananas

var bananaamount = document.getElementById("bananaamount");
bananacounter = bananacounter - 50;
bananaamount.textContent = bananacounter + " :";

}
else{

//Minus
svgContainer.append("text").attr("id", "moneytext")
.attr("x", 160)
.attr("y", 100)
.text(" - " + $amount)
.attr("font-family", "Chalkboard")
.attr("font-size", "50px")
.attr("fill", "red");

var moneyamount = document.getElementById("moneyamount");
if(moneycounter > 0){
moneycounter = moneycounter - $amount;
}
}

moneyamount.textContent = moneycounter + " :";
entered = true;
}

/*=====
Function that shows the sources
=====*/
function showsources(){
$("#characterselection").hide();
$("#sourcediv").show();
}

function backfromsource(){
$("#sourcediv").hide();
$("#characterselection").show();
}

```

```

}

/*=====
Function for the Game rules
=====*/
function showgamerules(){
$("#characterselection").hide();
$("#gamerules").show();
}

function backfromgamerules(){
$("#gamerules").hide();
$("#characterselection").show();
}

/*=====
Function that is called by the Espalda Button
=====*/
function espalda_clicked(){
if(dicethrown == false){
if (confirm('De verdad quieres volver a la pantalla de inicio?')) {
    $("#maingamesvg").hide();
    initialise_levels_selectionscreen();
    $("#characterselection").show();

//Play the sound corresponding to the level
switch(selected_level){

case "level1":
sound_backgroundmusic_level1.pause();
break;
case "level2":
sound_level2.pause();
break;
case "level3":
sound_level3.pause();
break;

} // End Switch Case

sound_intromusic.play();

reset_maingamesvg();

}
}
}

/*=====
Function that resets the maingamesvg
=====*/
function reset_maingamesvg(){

//Clear all images from maingamesvg
$("#maingamesvg").empty();

//Reset the necessary variables
banacounter = 0;
moneycounter = 0;
currentfield = 1;
entered = false;
deletearrow = true;
rounds_played = 1;
number_correct_answered = 0;
number_wrong_answered = 0;

```

```
}

/*=====
Function that resets the saved values
=====*/
function reset_saved_score(){

  if (confirm('Tu desea restablecer las puntuaciones')) {

    localStorage.removeItem("level");

    localStorage.removeItem("correct_level1");
    localStorage.removeItem("wrong_level1");

    localStorage.removeItem("correct_level2");
    localStorage.removeItem("wrong_level2");

    localStorage.removeItem("correct_level3");
    localStorage.removeItem("wrong_level3");

    localStorage.removeItem("character");      //Unlock the character

    initialise_levels_selectionscreen();

  }
}
```

Anhang IV – Quellcode Schlossspiel

Dies ist nur ein Ausschnitt des Codes. Der gesamte Code ist unter folgender Adresse abrufbar.
<https://github.com/kreuzerk/AdventuraMathematica>

```

var castlegame_characterposition = 0;           //Set the characterposition at the
beginning to zero
var castlegame_result = new Array(3); //Here we store the three results
var castlegame_liveplayer;
var castlegame_livecastle;

/*=====
    Reset lives
=====*/
function castlegame_resetlives(){
castlegame_liveplayer = 3;
castlegame_livecastle = 3;
//Reset the lives on the display
document.getElementById("livecastle").innerHTML = "Castillo: 3";
document.getElementById("liveplayer").innerHTML = "Tu: 3";
}

/*=====
    Create castlegame
=====*/
function castlegame_initialise(){

document.getElementById("castlegame_maincharacter").setAttribute("src", filename);

//Hide the fireballs from the castle and from the player
$( ".fireball" ).hide();
$( ".fireball_player1" ).hide();
$( ".fireball_player2" ).hide();
$( ".fireball_player3" ).hide();

create_castleexercise();

}

/*=====
    Create castleexercise
=====*/
function create_castleexercise(){

for(var i = 0; i<3; i++){

do{
var number1 = Math.floor(Math.random() * 20) + 1;
var number2 = Math.floor(Math.random() * 20) + 1;

var ok = true;

if(number1 % number2 != 0)
ok = false;

for(var j=0; j<castlegame_result.length; j++){
if(castlegame_result[j] == operators[level_operator](number1,number2)){ //Call the operators
function declared in the selectionscreen.js
ok = false;
break;
}
}
}
}
}

```

```

}
while(!ok);

castlegame_result[i] = operators[level_operator](number1,number2);
document.getElementById("castlegame_choice" + (i+1)).innerHTML = number1 + " " +
level_operator + " " + number2;

}

//Set the result to the paper at the top
document.getElementById("paperh1").innerHTML = castlegame_result[Math.floor(Math.random() *
3)];

}

//End of method create_castleexercise

/*=====
   This function is called by the control button
=====*/
function castlegame_controll(){

var position;

switch(castlegame_characterposition){
case -1:
position = 0;
break;

case 0:
position = 1;
break;

case 1:
position = 2;
break;

}

var papervalue = document.getElementById("paperh1").innerHTML;

if(castlegame_result[position] == papervalue){
castlegame_fire();    //Call the method to fire on the castle
}
else
{
castlegame_castle_fire();
}

}

/*=====
   This function is called by the left button
=====*/
function castlegame_go_left($id){

castlegame_characterposition--;    //Dekrement the castlegame_characterposition

if(castlegame_characterposition == -1 || castlegame_characterposition == 0){
sound_woosh.play();
$( ".block" ).animate({ "left": "-=240px" }, "slow" );
}
else{

```

```

castlegame_characterposition = -1; //Provend that the character goes to far
}
}

function castlegame_go_right(){

castlegame_characterposition++; //Inkrement the castlegame_characterposition

if(castlegame_characterposition == 1 || castlegame_characterposition == 0){
sound_woosh.play();
$( ".block" ).animate({ "left": "+=240px" }, "slow" );
}
else{
castlegame_characterposition = 1; //Provend that the character goes to far
}
}

/*=====
   This function is called when the castle fires
=====*/
function castlegame_castle_fire(){

sound_gunshoot.play();
$( ".fireball" ).show();

var left;

//Fire depending on the position of the caracter
switch(castlegame_characterposition){

case -1:
left = "900";
break;
case 0:
left = "660";
break;
case 1:
left = "420";
break;
}

$( ".fireball" ).animate({ "left": "-=" + left + "px", "top": "+=500px" }, "slow",
function(){
    $(".fireball").hide();
    castlegame_liveplayer--; //Dekrement the life of the player
    document.getElementById("liveplayer").innerHTML = "Tu: " + castlegame_liveplayer;
    $( ".fireball" ).animate({ "left": "+=" + left + "px", "top": "-=500px" }); //Reset the
    fireball to the original position
    sound_bomb.play();
    $( ".block" ).effect( "pulsate" ); //Pulsate effect for the .block class
    if(castlegame_liveplayer == 0){
    checksilver(false); //Call the checksilver with false. This means the player will
    lose. This method is located in the mainscript.js
    }
    else{
    castlegame_initialise();
    }
    });
}

/*=====
   This function is called when the player fires
=====*/
function castlegame_fire(){

```



```

var left;
var id;

sound_gunshoot.play();

switch(castlegame_characterposition){
case -1:
left = "850";
id="fireball_player1";
break;
case 0:
left = "610";
id="fireball_player2";
break;
case 1:
left = "370";
id="fireball_player3";
break;
}

$(". " + id).show();
$( " " + id ).animate({ "left": "+=" + left + "px", "top": "-=300px" }, "slow", function(){
    $(". " + id).hide();
    castlegame_livecastle--;
    document.getElementById("livecastle").innerHTML = "Castillo: " + castlegame_livecastle;
    $( " " + id ).animate({ "left": "-=" + left + "px", "top": "+=300px" }); //Reset the
    fireball to the original position
    sound_bomb.play();
    $( "#castleimage" ).effect( "pulsate" );

    if(castlegame_livecastle == 0){
    checksilver(true); //Call the check silver method. This method is located in the
    mainscript.js
    }
    else{
    castlegame_initialise();
    }
});
}

```

Anhang V – Quellcode Memoryspiel

Dies ist nur ein Ausschnitt des Codes. Der gesamte Code ist unter folgender Adresse abrufbar.
<https://github.com/kreuzerk/AdventuraMathematica>

```

var numberrange;
var memorygame_exercises = new Array(8);
var memorygame_results = new Array(8);
var memorygame_display_array = new Array(16);
var memorygame_result_array = new Array(16);

//Variables to store the element that has been clicked before
var memorygame_firstclicked = null;
var memorygame_firstclicked_id;
var memorygame_secondclicked_id = null;
var memorygame_previous_correct;
var previous_correct = false;
var memorygame_found_counter = 0;
var memorygame_numberoftries = 0;

/*=====
Function that initialises the memorygame
=====*/
function initialise_memory_game(){

memorygame_reset(); //Reset the memorygame to create a new exercise

//Set a different numberrange depending on the leveloperator
if(level_operator == "+" || level_operator == "-"){
numberrange = 50;
}
else{
numberrange = 10;
}

for(var i = 0; i<8; i++){

var numberisok = true;

do{ //Because of the division we only want numbers that have no rest on a division
var number1 = Math.floor(Math.random() * numberrange) + 1;;
var number2 = Math.floor(Math.random() * numberrange) + 1;;

for(var j = 0; j<memorygame_exercises; j++){ //Garante that the number is unique
if(memorygame_exercises[j] == number1 + " " + level_operator + " " + number2){
numberisok = false;
}
else{
numberisok = true;
}
}
} //End for loop

if(number1 % number2 != 0){ //Garante that the result of the division of the two numbers is
zero
numberisok = false;
}
else{
numberisok = true;
}
}
while(!numberisok);

memorygame_exercises[i] = number1 + " " + level_operator + " " + number2;

```

```

memorygame_results[i] = operators[level_operator](number1, number2);           //Store the
result
} //End for loop

initialise_display_result_array();
}

/*=====
Function that resets the whole application
=====*/
function memorygame_reset(){

//Reset the image and the onclick function
for(var i = 1; i<=16; i++){
document.getElementById("memorygame_td" + i).style.backgroundImage =
'url(/images/memorygame/memorycard_frontside.png)';
//Set the onclick method
document.getElementById("memorygame_td" + i).setAttribute("onclick",
"memorycard_clicked($(this).attr('order'), $(this).attr('id'))");
document.getElementById("memorygame_td" + i).innerHTML = "";
memorygame_found_counter = 0;
memorygame_numberoftries = 0;
document.getElementById("memorygame_intentos").innerHTML = "Intiendos: " +
memorygame_numberoftries;
}

}

/*=====
Function that initialises and creates the memorygame_display_array and the
memorygame_result_array
=====*/
function initialise_display_result_array(){

//Fill the memorygame_display_array
for(var i = 0; i<memorygame_display_array.length; i++){

if(i <= 7){
memorygame_display_array[i] = memorygame_exercises[i];
}
else{
memorygame_display_array[i] = memorygame_results[i-8];
}
} //End for loop

//Fill the memorygame_result_array
for(var i = 0; i<2; i++){

for(var j = 0; j<8; j++){
memorygame_result_array[(i*8)+j] = memorygame_results[j];
} //End first for

} //End second for

randomize_memorygame_arrays();

}

/*=====
Function that randomizes the memory_display and the memory_result_array
=====*/
function randomize_memorygame_arrays(){

for(var i = 0; i<50; i++){

```

```

var randomposition1 = Math.floor(Math.random() * memorygame_display_array.length);
var randomposition2 = Math.floor(Math.random() * memorygame_display_array.length);

//Create temporary variable and then switch the positions
var temp;

temp = memorygame_display_array[randomposition1];
memorygame_display_array[randomposition1] = memorygame_display_array[randomposition2];
memorygame_display_array[randomposition2] = temp;

//Also switch the result array accordingly to the other array
temp = memorygame_result_array[randomposition1];
memorygame_result_array[randomposition1] = memorygame_result_array[randomposition2];
memorygame_result_array[randomposition2] = temp;
}

}

/*=====
Function that is called if a memorycard is clicked
=====*/
function memorycard_clicked($order, $id){

$( "#" + $id ).hide();
$( "#" + $id ).show( "slide");

document.getElementById($id).style.backgroundImage =
'url(/images/memorygame/memorycard_backside.png)';
document.getElementById($id).innerHTML = memorygame_display_array[$order-1];

//Check if the first and a second element were clicked
if(memorygame_firstclicked != null && memorygame_secondclicked_id != null){

if(previous_correct == false){
document.getElementById(memorygame_firstclicked_id).innerHTML = "";
document.getElementById(memorygame_secondclicked_id).innerHTML = "";

//Reset the images to the front card_side
document.getElementById(memorygame_firstclicked_id).style.backgroundImage =
'url(/images/memorygame/memorycard_frontside.png)';
document.getElementById(memorygame_secondclicked_id).style.backgroundImage =
'url(/images/memorygame/memorycard_frontside.png)';

document.getElementById(memorygame_firstclicked_id).setAttribute("onclick",
"memorycard_clicked($(this).attr('order'), $(this).attr('id'))");
document.getElementById(memorygame_secondclicked_id).setAttribute("onclick",
"memorycard_clicked($(this).attr('order'), $(this).attr('id'))");

}
memorygame_firstclicked = null; //Reset the first clicked element to null
memorygame_secondclicked_id = null; //Reset the second element to null

}

//Needs to be before the second if statement otherwise it does not work
if(memorygame_firstclicked != null){

memorygame_numberoftries++; //Inkrement the number of tries
document.getElementById("memorygame_intentos").innerHTML = "Intiendos: " +
memorygame_numberoftries;

//This if is executed if the code is correct
if(memorygame_result_array[memorygame_firstclicked-1] == memorygame_result_array[$order-1]){
memorygame_found_counter++;

```

```

if(memorygame_found_counter < 8){

document.getElementById(memorygame_firstclicked_id).style.backgroundImage =
'url(../images/memorygame/Imageok.png)';
document.getElementById($id).style.backgroundImage = 'url(../images/memorygame/Imageok.png)';
document.getElementById(memorygame_firstclicked_id).setAttribute("onclick", "");
document.getElementById($id).setAttribute("onclick", "");

previous_correct = true;
}
else
{
//Call the function that is responsible for checking this fields

if(memorygame_numberoftries <= 30){
    checkchocolate(true, 10);
}
else{
    if(memorygame_numberoftries <= 40){
        checkchocolate(true, 0);    //Call the checkchocolate located in the
mainscript.js
    }
    else
    {
        checkchocolate(false,10);    //Call the checkchocolate located in the
mainscript.js
    }
}

}

}
else
{
previous_correct = false;
}
memorygame_secondclicked_id = $id;

}
else{
memorygame_firstclicked = $order;
memorygame_firstclicked_id = $id;
document.getElementById($id).setAttribute("onclick", "");
}

}

```

Anhang VI – Quellcode Quizspiel

Dies ist nur ein Ausschnitt des Codes. Der gesamte Code ist unter folgender Adresse abrufbar.
<https://github.com/kreuzerk/AdventuraMathematica>

```

var quizgame_result;
var quizgame_tigerpoints;
var quizgame_tupoints;

/*=====
Function that resets the points of the game
=====*/
function quizgame_reset_points(){
//Set the points at the beginning of the game to zero
quizgame_tigerpoints = 0;
quizgame_tupoints = 0;
}

/*=====
Function that initialises the quizgame
=====*/
function initialise_quizgame(){

$("#quizgame_tigeranswer").hide();
$("#quizgame_tuanswer").hide();

//Show the input field and the ok button
$(".quiz_answerdiv").show();
document.getElementById("quiz_input").value = "";

document.getElementById("quizgame_tiger_pointdisplay").innerHTML = "Tigre: " +
quizgame_tigerpoints;
document.getElementById("quizgame_tu_pointdisplay").innerHTML = "Tu: " + quizgame_tupoints;

quizgame_createexercise();
}

/*=====
Function that is responsible for creating the exercise
=====*/
function quizgame_createexercise(){

do{
var number1 = Math.floor(Math.random() * 12) + 1;
var number2 = Math.floor(Math.random() * 12) + 1;
}
while(number1 % number2 != 0);    //Make sure that the result of a division is a whole
number

document.getElementById("quiz_exercise_responsa_correcta").innerHTML = "Responde esta
pregunta";
document.getElementById("quiz_exercise_h1").innerHTML = number1 + " " + level_operator + " " +
number2;
quizgame_result = operators[level_operator](number1,number2);
}

/*=====
Function that is called if the ok button is clicked
=====*/
function quizgame_ok_clicked(){

sound_blop.play();

```

```

$(".quiz_answerdiv").hide();
window.setTimeout(quizgame_showresults,1000);

}

/*=====
Function that is responsible for showing the results
=====*/
function quizgame_showresults($number){

var number = document.getElementById("quiz_input").value;
var tigernumber = tiger_is_calculating(); //Call the method tiger is calculating and store
the return value

document.getElementById("quizgame_tuanswer_h1").innerHTML = number;
document.getElementById("quizgame_tigeranswer_h1").innerHTML = tigernumber;

document.getElementById("quiz_exercise_responsa_correcta").innerHTML = "";
document.getElementById("quiz_exercise_h1").innerHTML = "";

$("#quizgame_tigeranswer").show();
$("#quizgame_tuanswer").show();

window.setTimeout(quizgame_display_correct_answer,2000);
}

/*=====
Function that represents the intelligence of the tiger
=====*/
function tiger_is_calculating(){

// Returns a random number between min (inclusive) and max (exclusive)

var max = quizgame_result + 4;
var min = quizgame_result - 4;

for(var i = 0; i<5; i++){
var number = Math.floor(Math.random() * (max - min + 1)) + min;

if(number == quizgame_result){
break;
}
}

return number;
}

/*=====
Function that displays the correct answer
=====*/
function quizgame_display_correct_answer(){

document.getElementById("quiz_exercise_responsa_correcta").innerHTML = "La respuesta correcta
es";
document.getElementById("quiz_exercise_h1").innerHTML = quizgame_result;

//Display the number of received points inside of the bubble
var tigeranswer = document.getElementById("quizgame_tigeranswer_h1").innerHTML;

if(tigeranswer == quizgame_result){ //Tiger has answered correctly

document.getElementById("quizgame_tigeranswer_h1").innerHTML = "+ 1";
sound_tigerlaugh.play();

```

```

quizgame_tigerpoints++;
document.getElementById("quizgame_tiger_pointdisplay").innerHTML = "Tigre: " +
quizgame_tigerpoints;
}
else{
document.getElementById("quizgame_tigeranswer_h1").innerHTML = "Mal";
}

var tuanswer = document.getElementById("quizgame_tuanswer_h1").innerHTML;

if(tuanswer == quizgame_result){ //The player has answered correctly
document.getElementById("quizgame_tuanswer_h1").innerHTML = "+ 1";
sound_bananaup.play();
quizgame_tupoints++;
document.getElementById("quizgame_tu_pointsdisplay").innerHTML = "Tu: " + quizgame_tupoints;
}
else{
document.getElementById("quizgame_tuanswer_h1").innerHTML = "Mal";
}

//Reinitialise the quizgame
if(quizgame_tigerpoints < 6 && quizgame_tupoints < 6){
window.setTimeout(initialise_quizgame,3000);
}
else
{

if(quizgame_tigerpoints == 6 && quizgame_tupoints == 6){
quizgame_finished(0); //0 = Draw
}
else if(quizgame_tigerpoints == 6){
quizgame_finished(-1); //-1 = Tiger wins
}
else{
quizgame_finished(1); // 1 = Player wins
}

//Pause and play the sounds
sound_tigerbackground2.pause();
sound_tigerbackground1.play();
sound_tigerlaugh.play();

}

}

```


Anhang VII – Quellcode „Was kommt als nächstes?“

Dies ist nur ein Ausschnitt des Codes. Der gesamte Code ist unter folgender Adresse abrufbar.
<https://github.com/kreuzerk/AdventuraMathematica>

```
var pattern1 = ["rectblue", "circleyellow", "circlegreen", "rectblue", "circleyellow",
"circlegreen", "rectblue", "circleyellow", "circlegreen", "rectblue"];
var pattern2 = ["circleblue", "circleyellow", "circlegreen", "circleblue", "circleyellow",
"circlegreen", "circleblue", "circleyellow", "circlegreen", "circleblue"];
var pattern3 = ["rectblue", "rectblue", "circleyellow", "circlegreen", "rectblue", "rectblue",
"circleyellow", "circlegreen", "rectblue", "rectblue"];
var pattern4 = ["rectgreen", "circleblue", "rectgreen", "circleblue", "rectgreen",
"circleblue", "rectgreen", "circleblue", "rectgreen", "circleblue"];
var pattern5 = ["circleyellow", "circleblue", "circlegreen", "circleyellow", "rectblue",
"circleyellow", "circleblue", "circlegreen", "circleyellow", "rectblue"];
var pattern6 = ["rectblue", "circlegreen", "circleyellow", "rectgreen", "rectblue",
"circlegreen", "circleyellow", "rectgreen", "rectblue", "circlegreen"];
var pattern7 = ["circleyellow", "circlegreen", "rectblue", "circleyellow", "circlegreen",
"rectblue", "circleyellow", "circlegreen", "rectblue", "circleyellow"];
var pattern8 = ["circleblue", "circlegreen", "rectyellow", "circleyellow", "circleblue",
"circlegreen", "rectyellow", "circleyellow", "circleblue", "circlegreen"];

var solution = new Array(10);

var pattern;
var pattern_drawn;
var numberofpatterns = 8;

var pattern1_drawn = 4;
var pattern2_drawn = 4;
var pattern3_drawn = 5;
var pattern4_drawn = 3;
var pattern5_drawn = 4;
var pattern6_drawn = 5;
var pattern7_drawn = 4;
var pattern8_drawn = 5;

var patternnumber;

var next_x = -80;
var next_y = 140;
var aftercircle = false;
var drawncounter = -1;
var colors = ["blue", "yellow", "green"];

/*=====
Function that draws the nextsvg exercise
=====*/
function nextsvg_drawexercise(){

//Clean the svg element
cleannextgamesvg();

//Choose the pattern
patternnumber = Math.floor(Math.random() * numberofpatterns) + 1;
nextsvg_drawelements();
}

function nextsvg_drawelements(){

//Draw the text
nextsvg.append("text").
    attr("x", 20)
```

```

        .attr("y", 40)
        .text("Cuenta el número de veces que aparece cada forma")
        .attr("font-family", "Chalkboard")
        .attr("font-size", "30px")
        .attr("fill", "black");

selectsvg.append("text").
    attr("x", 20)
    .attr("y", 40)
    .text("Cuenta el número de veces que aparece cada forma")
    .attr("font-family", "Chalkboard")
    .attr("font-size", "30px")
    .attr("fill", "black");

switch(patternnumber){
case 1:
    pattern = pattern1;
    pattern_drawn = pattern1_drawn;
break;
case 2:
    pattern = pattern2;
    pattern_drawn = pattern2_drawn;
break;
case 3:
    pattern = pattern3;
    pattern_drawn = pattern3_drawn;
break;
case 4:
    pattern = pattern4;
    pattern_drawn = pattern4_drawn;
break;
case 5:
    pattern = pattern5;
    pattern_drawn = pattern5_drawn;
break;
case 6:
    pattern = pattern6;
    pattern_drawn = pattern6_drawn;
break;
case 7:
    pattern = pattern7;
    pattern_drawn = pattern7_drawn;
break;
case 8:
    pattern = pattern8;
    pattern_drawn = pattern8_drawn;
break;
}

//Drawing the exercise part
//=====

next_x = -60;
next_y = 120;

var drawingcolor;

//Go through all the elements of the selected pattern and get its color and draw the elements
with the color
for(var i = 0; i < pattern_drawn; i++){

if(pattern[i].indexOf("blue") >= 0){
    drawingcolor = "blue";

```

```

}

if(pattern[i].indexOf("yellow") >= 0){
    drawingcolor = "yellow";
}

if(pattern[i].indexOf("green") >= 0){
    drawingcolor = "green";
}

if(pattern[i].indexOf("rect") >= 0){
    drawnnextelement(drawingcolor, true);
}
else
{
    drawnnextelement(drawingcolor, false);
}

}

}

/*=====
Drawing the Selection part
=====*/
function nextsvg_drawselections(){
var x = 85;
var y = 150;

//Draw the rectangles
for(var i = 0; i < colors.length; i++){

    selectsvg.append("rect").attr("onclick",
"drawfigure($(this).attr('form'),$(this).attr('fill'))")
                .attr("x", x)
                .attr("y", y)
                .attr("width", 100)
                .attr("height", 100).attr("form", "rect")
                .attr("fill", colors[i]).attr("stroke","black");

    x = x + 185;
}

x = 690;
y = 200

//Draw the circles
for(var i = 0; i < colors.length; i++){

    selectsvg.append("circle").attr("onclick",
"drawfigure($(this).attr('form'),$(this).attr('fill'))")
                .attr("cx", x)
                .attr("cy", y)
                .attr("r", 50).attr("form","circle")
                .attr("fill", colors[i]).attr("stroke","black");

    x = x + 185;
}
} //End of function nextsvg_drawselections

/*=====
=====
Function for drawing the next element
=====

```

```

*/
function drawnnextelement($drawingcolor, $rect){

drawncounter++;          //Checks how many elements are drawn

if($rect){
next_y = 120; //If it is a rect we need another y coordinate

if(aftercircle == false){
next_x = next_x + 116; //If the next elements comes after a rect we need another x coordinate
}
else
{
next_x = next_x + 76;
}

//Draw the rect
nextsvg.append("rect")
                .attr("x", next_x)
                .attr("y", next_y)
                .attr("width", 80)
                .attr("height", 80).attr("form", "rect")
                .attr("fill", $drawingcolor).attr("stroke", "black");

//Add the drawn form to the solution array
solution[drawncounter] = "rect" + $drawingcolor;

//Say that the next form is not after a circle
aftercircle = false;

}
else
{

next_y = 160;

if(aftercircle == false){
next_x = next_x + 156;          //If it is after a rectangle we need another x coordinate
}
else{
next_x = next_x + 116;
}

//Draw the circle
nextsvg.append("circle")
                .attr("cx", next_x)
                .attr("cy", next_y)
                .attr("r", 40).attr("form", "circle")
                .attr("fill", $drawingcolor).attr("stroke", "black");

//Add the drawn form to the solution
solution[drawncounter] = "circle" + $drawingcolor;

aftercircle = true;          //Say that the next form is drawn after a circle
}

//Do the correction of the exercise

if(drawncounter == 9){          //9 because it begins with zero at the first iteration

var formiscorrect = true;

//Check if everything is correct

```

```

    for(var i=0; i < solution.length; i++){
        if(solution[i] != pattern[i]){
            formiscorrect = false;
            break;
        }
    }

    checkpinkfields(formiscorrect);
}

}

/*=====
Function for handling the click on an element
=====*/
function drawfigure($form, $color){

    sound_pencil.play();

    if($form == "rect"){
        drawnextelement($color, true);
    }
    else
    {
        drawnextelement($color, false);
    }

}

/*=====
Function to remove everything from the first svg
=====*/
function cleannextgamesvg(){
    drawncounter = -1;
    $("#nextgamesvg").empty();
}

/*=====
Function that is called if the reset button is clicked
=====*/
function correct_nextgame(){

    cleannextgamesvg(); //Clean the elements
    nextsvg_drawelements(); //Redraw the exercise

}

```

32 Selbständigkeitserklärung

Ich bestätige hiermit, dass ich die vorliegende Bachelorarbeit alleine und nur mit den angegebenen Hilfsmitteln realisiert habe und ausschliesslich die erwähnten Quellen benutzt habe. Ohne Einverständnis des Studiengangsleiters David Wannier und des für die Bachelorarbeit verantwortlichen Dozierenden Laurent Sciboz werde ich diesen Bericht an niemanden verteilen.

Siders, Juli 2014

Kevin Kreuzer